

A Framework for Generating Informative Answers for Question Answering Systems



Arangalage Rivindu Prasanga Perera

School of Engineering, Computer and Mathematical Sciences

Auckland University of Technology

A thesis submitted to Auckland University of Technology in fulfilment
of the requirements for the degree of
Doctor of Philosophy

September 2017

To all those who seek knowledge

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

A handwritten signature in blue ink that reads "Rivindu". The signature is written in a cursive style and is underlined.

Arangalage Rivindu Prasanga Perera

September 2017

Acknowledgements

I am extremely grateful to the many people who have helped me in the study for and preparation of the work described in this thesis. First of all, I would like to thank my parents, Nimal Perera and Chandrani Denawaka, and sister, Chathuranga Perera, who supported me in all my pursuits and encouraged my academic interests from day one.

I would like to thank my primary supervisor, Dr. Parma Nand, for his excellent guidance, advice, and continuous encouragement throughout this period of three years. During this period, he contributed to a rewarding research experience by giving me enough intellectual freedom in my work, supporting my attendance at various conferences, arranging research fundings, co-authoring 28 research papers, and demanding a high quality of work in all my endeavours.

I am also grateful to Prof. Wai (Albert) Yeap, co-supervisor of this research, for his guidance and arranging all the facilities at Centre for Artificial Intelligence Research (CAIR).

Completing this research would have been all the more difficult were it not for the support received from Auckland University of Technology (AUT). I would like to acknowledge the School of Engineering, Computer and Mathematical Sciences (SECMS) and Faculty of Design and Creative Technologies for funding my research with three scholarship schemes which covered both the fees and living expenses. Special thanks go to Karishma Bhat, Saide Lo, and Terry Brydon; Karishma handled administrative side of the PhD with her great commitment, Saide organized all my visits to conferences with

her amazing organizational skills, and Terry accepted all my requests to access various resources with a smile. I am also thankful to Kerri Spooner, Dr. Boris Bacic, and Assoc. Prof. Wei Qi Yan, for the support, advice, and consistent encouragement that I received throughout last three years. I am also grateful to Bumjun Kim and the technical support team for helping me to fix various technical problems. Thank you also to the Software Engineering Research Lab (SERL) staff for their consistent encouragement and support. My sincere thanks also goes to Ruby Roebuck for awesome proofreading she did to improve the thesis. I would also like to thank all the academic and non-academic staff at AUT for helping me in various occasions and for nice discussions we had.

Lastly, I would like to thank some of my research collaborators who worked with me during last three years and helped me to improve this research in various ways. Thank you Prof. Kazuhiro Seki, Dr. Radek Burget, Dr. Kristin Stock, Prof. Kohichi Toshioka, and Dr. Wen-Hsin Yang, for all your comments and words of encouragement.

Abstract

Recent trends in Question Answering (QA) systems have led to a proliferation of studies which have focused on building advanced QA systems which are able to compete with the QA ability of humans. To this effect, a large number of these systems have shifted to Question Answering over Linked Data (QALD). The use of Linked Data as the basis of knowledge representation by the QA systems has led to noticeable improvements in both recall and precision compared to the conventional, unstructured text based systems. However, answers from these systems are still not able to mimic human generated answers, which has been an ambition for Artificial Intelligence (AI) researchers for more than a decade. One of the two main reasons for the “machine feel” of the answers has been the inability of QA systems to present the answer as a fully constructed, natural language sentence. The second reason is that humans generally answer a question with elaboration containing additional contextual information, apart from the specific answer to the question. This aspect has been especially challenging for QA systems as it is difficult to source the contextual information, rank them and then formulate the information as multiple sentences in a form that emanates human generated text.

Previous research has investigated answer presentation by summarizing unstructured text, selecting contextual information from a closed domain ontology, and using cooperative and user tailored answers, however, these studies have not dealt with the generation of an answer in natural language with additional contextual information.

This thesis describes a framework, *RealText*, which presents an answer from a QA system in a natural language form, together with extraneous contextual information. This answer, referred hereafter as informative answer, comprises a sentence which presents the answer as a natural language sentence, and in addition, contains an elaboration of the entities contained in both the question and the answer.

The information required to generate the elaborations was retrieved from DBpedia, which is an open domain Linked Data resource and is considered to be the nucleus for the ever-growing Linked Data cloud. Linked Data is represented in a structured form as a triple, and this enables the required information to be selected for the identified entities with no ambiguity compared to the use of unstructured text summarization which is prone to a high level of ambiguity. With the current rate of growth, Linked Data is set to become much more prevalent which will mean development of a lot more of Linked Data resources getting linked to DBpedia, thus making it a central hub for the Linked Data cloud. This would put architectures that use DBpedia as the knowledge source at an advantage.

The generation of an elaboration paragraph based on structured information contained in the triples requires several steps. The triples firstly need to be lexicalized, which involves transformation of the individual triples into basic stand-alone sentences. These sentences then need to be further enhanced and meshed into a paragraph using linguistic tasks such as aggregation and referral expression generation. The RealText framework integrates these linguistic processes as used by humans to generate a paragraph consisting of multiple sentences. Additionally, the framework implements realization functions and inferences on gender, ontology class of an entity, inter alia, to further enhance the text to make it more akin to human generated text.

We used the QALD evaluation campaign dataset which contains the question, the query, and the answer as the source data. Since we were working in the final answer

presentation stage, extraction of the answer was out of the scope for this project. Additionally, the framework uses all of the triples associated with a given entity, hence does not focus on ranking triples. The list of triples used in the contextual information generation is provided by DBpedia which is the structured version of Wikipedia.

The evaluation of research of this nature is challenging for two reasons; firstly there is no benchmark data available and secondly evaluation of natural text can only be done accurately by human evaluators which is expensive, both in terms of money and time. We evaluated the RealText framework based on three criteria; readability, accuracy, and informativeness. Measurement of these criteria is highly subjective and difficult to measure as definite scientific variables. It is far more challenging to implement automated systems to measure these criteria. To validate this research, we principally used human participants to evaluate the “naturalness” of the generated text under a condition in which the inter-annotator agreement was computed to make sure that there was a minimum threshold of agreement between the participants. In addition, we also investigated several automated metrics to see if anyone of them could correlate with the human evaluations.

The results showed that more than 95% of the generated answers achieved an average rating above three out of five for all of the criteria. Furthermore, 39.02% of the generated answers achieved an average rating above four for the readability criteria, while the value for accuracy and informativeness were in the vicinity of 66%. Further, the investigation into the automated metrics showed that none of the metrics correlated with the human evaluations.

In summary, this thesis presents a framework that would be able to generate multi sentence “natural text” based on a given set of entities by extracting the information from a linked data knowledge base such as DBpedia. The framework being presented is robust enough to be able to generate text for any given set of entities, hence would be

extendable to any natural language generation task, such as description text generation for kiosks, dialogue systems for Intelligent Personal Assistants (IPA), patient summary generation in eHealth, and narrative generation in eLearning applications.

Table of Contents

List of Figures	xiv
List of Tables	xvii
List of Codes	xxi
Nomenclature	xxii
1 Introduction	1
1.1 Publications from this Research	7
1.1.1 Journal Publications	8
1.1.2 Conference Proceedings	8
1.1.3 Other Related Publications	11
2 Literature Review	12
2.1 Definitions	13
2.2 Answers with Contextual Information	14
2.2.1 Summarization Approach	16
2.2.2 Selectional Approach	23
2.3 Cooperative Answers	29
2.3.1 Intensional Answers	30
2.3.2 Alternative Queries and Query Relaxation	39

2.4	User Tailored Answers	44
2.5	Evaluating Answer Presentation	48
2.5.1	Human Evaluation	48
2.5.2	Automatic Metric based Evaluation	51
2.6	Chapter Summary	57
3	Methodology	59
3.1	RealText Architecture	59
3.2	DBpedia as an Information Source	64
3.2.1	Structure of DBpedia	64
3.2.2	DBpedia Suitability for QA Systems	66
3.2.3	DBpedia Databases	69
3.3	Language Resources	74
3.3.1	Verb Information Database	74
3.3.2	Masculine-Feminine Token Database	74
3.4	Question Dataset	76
3.4.1	Entity Extraction	76
3.4.2	Triple Extraction and Metadata Embedding	78
3.4.3	Answer Type Classification and Verbalization	79
3.4.4	Structure of the Enhanced Question Dataset	80
3.5	Answer Sentence Generation	80
3.5.1	Answer Sentence and Process Overview	82
3.5.2	Question Type Identification	83
3.5.3	Pattern Extraction from Dependency Tree	84
3.5.4	Pattern Search	88
3.5.5	Answer Merging	88
3.5.6	Sentence Realization	92

3.6	Lexicalization	92
3.6.1	Lexicalization High-level Architecture	96
3.6.2	Occupational Metonym Patterns	98
3.6.3	Context Free Grammar Patterns	101
3.6.4	Relational Patterns	102
3.6.5	Property Patterns	113
3.6.6	Pattern Search and Realization	115
3.6.7	Output of the Lexicalization	124
3.7	Aggregation	125
3.7.1	Overview of the Aggregation Process	125
3.7.2	Subject based Clustering	126
3.7.3	Rule based Sub-Clustering and Aggregation	127
3.7.4	Further Realizing the Aggregation	133
3.7.5	Output from the Aggregation	135
3.8	Referring Expression Generation	135
3.9	Structure Realization	137
3.9.1	Entity Description Ordering	137
3.9.2	Presentation Formats	138
3.10	Chapter Summary	143
4	Evaluation	144
4.1	The Test Dataset	145
4.2	Module wise Evaluation and Statistics	148
4.2.1	Answer Sentence Generation	148
4.2.2	Lexicalization	153
4.2.3	Aggregation	174
4.2.4	Referring Expression Generation	177

4.2.5	Structure Realization	181
4.3	Human Evaluation Results	182
4.3.1	Evaluation Process	182
4.3.2	Results of the Human Ranking	186
4.3.3	Post hoc Analysis	188
4.4	Automatic Metric based Evaluation: An Investigation	191
4.4.1	Evaluation Process	191
4.4.2	Results of the Automatic Evaluation	193
4.4.3	Post hoc Analysis	198
4.5	Some Comparisons with Examples	204
4.6	Limitations and Assumptions	208
4.7	Chapter Summary	211
5	Conclusion	212
5.1	The Contributions of this Research	212
5.2	Future Works	215
5.3	Concluding Remarks	218
	References	220
	Appendix A Sample Test Question Results	229
	Index	240

List of Figures

1.1	An example scenario depicting the triples associated with two linked entities	7
2.1	Taxonomy of answer presentation research	12
2.2	Question classification schemes	13
2.3	An example of an annotated text segment with rhetorical relations . .	17
2.4	Example answer from AskHERMES system	21
2.5	AKT ontology and the answer generated by the AQUA QA system . .	24
2.6	Answer generated from START QA system	25
2.7	HITIQA data frame generation	27
2.8	An example answer from ORAKLE QA system	37
2.9	Two examples depicts the Meteor phrase matching technique	57
3.1	High-level view of the RealText architecture	60
3.2	Example of an informative answer	63
3.3	DBpedia ontology class visualization	65
3.4	Linked Open Data diagram	68
3.5	Attribute value matrix for an enhanced question	81
3.6	Schematic representation of the answer sentence generation process .	82
3.7	Dependency grammar relations between tokens in a question	85

3.8	Lexicalization high-level architecture	96
3.9	Classification hierarchy of English morphology	98
3.10	Two different occupational metonym formation applying -er nominals	99
3.11	Relational pattern extraction process	102
3.12	Example scenario of extracting a lexicalization pattern	110
3.13	Typed dependency parse to identify compound tokens	111
3.14	POS tagged transformed sentence	111
3.15	Dependency parse result of the type-1 gender specific pattern	119
3.16	Dependency parse result of the type-2 gender specific pattern	120
3.17	Output of the lexicalization module depicted in an attribute-value matrix	125
3.18	Overview of the aggregation architecture	126
3.19	A sample output from the aggregation module	136
4.1	Question wise classification of triples	147
4.2	Question in development set which contains seven dependency relations	150
4.3	Coverage of the extracted patterns in the test dataset	151
4.4	Q-Q plot the relation score and alignment score data	158
4.5	Statistics of the property pattern lexicon categorized based on the pattern type	162
4.6	Question wise lexicalization pattern type distribution	166
4.7	Inference based pattern distribution	167
4.8	Property pattern type wise distribution	167
4.9	Question wise lexicalization accuracy	171
4.10	Type wise lexicalization pattern accuracy	173
4.11	Inference level pattern accuracy	174
4.12	Distribution of aggregations based on the rule	176
4.13	Aggregation rule distribution per question	178

4.14	All lexicalization versus the aggregated lexicalizations	179
4.15	Human ranking based evaluation survey	186
4.16	Human ranking based evaluation results for readability, accuracy, and informativeness	189
4.17	Classification of questions into five ranges based on the human ranking evaluation	190
4.18	Sample survey question to collect human answers	192
4.19	Results of the automatic evaluation for four metrics	194
4.20	BLEU metric results using unigrams to quadgrams	196
4.21	ROUGE metric results using unigrams to quadgrams	197
4.22	Complete alignment of human reference and system answer sentence .	200
4.23	Answer sentence alignment with different positions	201
4.24	Alignment with synonym matching	202
4.25	Alignment with phrase matching	202

List of Tables

1.1	Some examples of ambiguous question and answer pairs	3
1.2	An example output that was generated from our framework	6
2.1	Evaluation results of the MedQA system	20
2.2	Summary of the answer presentation approaches using summarizing the contextual information	22
2.3	Summary of the answer presentation approaches by selecting the con- textual information	28
2.4	Extensional and intensional answers for the question “which countries use the Euro?”	30
2.5	Intensional answer categories derived from analysing a corpus	33
2.6	Summary of the answer presentation approaches using intensional answers	38
2.7	Summary of the answer presentation approaches using alternative queries and query relaxation	43
2.8	Summary of the answer presentation approaches using user tailored answers	47
3.1	Comparison of DBpedia statistics with Freebase, Yago, Wikidata, and OpenCyc	67
3.2	DBpedia growth rate in last seven releases	67

3.3	Statistics on DBpedia interlinking	69
3.4	Sample set of records from the gender database	70
3.5	Sample set of records from the measurement unit database	71
3.6	Sample set of records from the ontology class – predicate database	71
3.7	Sample set of records from the ontology class – entity database	72
3.8	Sample set of records from the predicate – date database	73
3.9	Sample set of records from the predicate number database	73
3.10	Verb information database	75
3.11	Sample set of records from the masculine – feminine token database	76
3.12	Initial answer type classification	79
3.13	Interrogative types with examples and associated POS tags	84
3.14	Examples of dependency subtrees extracted from parsed questions	86
3.15	Extracting and ordering phrases based on the selected pattern	89
3.16	Rules applied in the answer merging for wh-interrogatives	91
3.17	Examples of answer sentence generation with realization	93
3.18	Example lexicalization patterns	95
3.19	Sample input to the relational pattern extraction module	103
3.20	Decision process for relation-triple alignment	109
3.21	Lexicalization pattern with their respective scores and occurrence counts.	113
3.22	Property patterns with examples	114
3.23	Examples of OWL CNL pattern based lexicalizations	117
3.24	Relational pattern realization for active person realization	122
3.25	A sample clustering and ordering of triples	127
3.26	Sample set of predicates that can be aggregated	131
3.27	Dependency rules to identify the token to be pluralized	134
4.1	Statistics of the test dataset	146

4.2	Statistics of the Valid Triples	148
4.3	Statistics on the answer sentence generation patterns	149
4.4	An example of four dependency path based answer sentence generation pattern	149
4.5	Answer sentence generation basic statistics	150
4.6	Top-5 Patterns in answer sentence generation	152
4.7	Active and passive form of the subtree patterns	152
4.8	Example question for which an answer sentence was not generated due to the absence of patterns	153
4.9	Selected set of records from the occupational metonym lexicon	154
4.10	Statistics of the relational pattern generation process	155
4.11	Statistics of the extracted relational patterns	156
4.12	Multivariate normality analysis	158
4.13	Examples where alignment score and relational scores have high diversity	159
4.14	A set of records from the property pattern lexicon.	161
4.15	Statistics of the property pattern lexicon.	162
4.16	Example lexicalizations from the test data	164
4.17	Examples of realizations	169
4.18	Some examples of reasons behind inaccurate lexicalization	172
4.19	Example aggregations retrieved from the test dataset	175
4.20	Referring expression generation statistics	180
4.21	Example referring expressions generated for the test dataset	181
4.22	Correlation analysis between readability, accuracy, and informativeness	191
4.23	Inter-metric correlation matrix for the automatic metrics	199
4.24	Statistics of the alignment	200

4.25	Correlation values between the human ratings for readability and automatic metric values	203
4.26	Comparison of sample answers from Bosma's (2005) approach and the RealText framework	205
4.27	RealText generated entity descriptions for a question which contains two similar entities	206
4.28	A comparison between the answers provided by AQUA (Vargas-Vera and Motta, 2004) and RealText	207
4.29	A comparison between the answers provided by Cimiano et al.'s (2008) approach and RealText	209

List of Codes

3.1	SPARQL query to retrieve gender of an entity	70
3.2	SPARQL query for the question “Who was the successor of John F. Kennedy?”	77
3.3	SPARQL algebraic expression of the query shown in Listing 3.2	77
3.4	Identifying answer type using XML schema definitions associated with SPARQL query result	80
3.5	SPARQL query for the question “How tall is Michael Jordan?”	92
3.6	SPARQL algebraic expression of the query shown in Listing 3.5	92
3.7	A naive approach to pluralize English words	134
3.8	SSML annotated answer for a question	139
3.9	XML formatted answer for a question	140
3.10	HTML formatted answer for a question	141
3.11	L ^A T _E X formatted answer for a question	141
3.12	RDF formatted answer for a question	142
4.1	A generated answer presented in SSML form	183
4.2	A generated answer presented in RDF form	184

Nomenclature

Special Notations

$\langle \rangle_{AG}$ Aggregated Sentence

$\langle \rangle_{LT}$ Lexicalized Triple

$\langle \rangle_L$ Lexicalization Pattern

$\langle \rangle_R$ Relation

$\langle \rangle_T$ Triple

Acronyms / Abbreviations

AI Artificial Intelligence

ARI Automated Readability Index

AVM Attribute Value Matrix

CFG Context Free Grammar

ClosedIE Closed Information Extraction

CNL Controlled Natural Language

ILP Inductive Logic Programming

IPA	Intelligent Personal Assistant
IT	Information Technology
LCS	Lexical Conceptual Structure
LCS	Longest Common Subsequence
LGG	Least General Generalization
MT	Machine Translation
NLG	Natural Language Generation
NLP	Natural Language Processing
ODF	Open Document Format
OpenIE	Open Information Extraction
OWL	Web Ontology Language
POJO	Plain Old Java Object
POM	Phrasal Overlap Measure
POS	Part Of Speech
QA	Question Answering
QALD	Question Answering over Linked Data
RDF	Resource Description Framework
REG	Referring Expression Generation
RST	Rhetorical Structure Theory

SPARQL SPARQL Protocol and RDF Query Language

SSML Speech Synthesis Markup Language

TFIDF Term Frequency - Inverse Document Frequency

WER Word Error Rate

Chapter 1

Introduction

HAL 9000: *“Sorry to interrupt the festivities, but we have a problem.”*

Bowman: *“What is it?”*

HAL 9000: *“I am having difficulty in maintaining contact with Earth. The trouble is in the AE-35 unit. My Fault Prediction Center reports that it may fail within seventy-two hours.”*

The excerpt above is from the “*2001: A Space Odyssey*”, a masterpiece by Arthur C. Clarke in which HAL 9000 (an extremely powerful Artificial Intelligence (AI) program) is having a conversation with a real human, Bowman, who is an astronaut. Note that in response to Bowman’s question, HAL 9000 does not only respond with just an answer, but also an elaboration of the central information contained in the answer. In addition, the response from HAL 9000 is in a natural language and as full sentences, rather than a table of fields which is typically the standard output from a machine. These two characteristics in the response from HAL 9000 form the basis for the motivation of the research being reported in this thesis.

The presentation of such an answer adds two main enhancements to Question Answering (QA) systems.

- Provides extraneous contextual information, usually of use to human consumers.
- The additional information enables the consumer to validate the answer.

The first of the two points is beneficial to a human consumer as they are constantly in search of knowledge for various reasons, therefore the availability of appropriately relevant information aligns with this need. It is a further advantage if the additional information is presented in natural language, rather than as a link which is the typical format for a machine to human interaction. Information presented as a link is distracting for the consumer and can be a disincentive to navigate to it unless the information in the link is crucial for the consumer. The second point, validation, is a verity of natural language use. Since natural languages are abundant in words with ambiguities, both implicit and explicit contextual knowledge is frequently used to resolve the ambiguities in order to validate the answer. In the case of QA systems, the question, the answer, or both may be ambiguous and could have more than one interpretation. Table 1.1 shows example question-answer pairs containing ambiguity to varying extents. In the first example, Michael Jordan could refer to the famous footballer as well as one of other 10 well known people including businessman, racing driver, baseball player, and actor. In the second and third example, not only the entities mentioned in the question are ambiguous, but also the entities in the answers. For instance, Apache may refer to TVS Apache motorbike, Apache helicopter, Apache Cooperation, Apache Web server, Apache Software Foundation, and Apache Film (Wikipedia shows Apache disambiguation of 40 instances categorized under 7 domains). Although, some of the entities (e.g., TVS Apache motorbike, Apache helicopter, and Apache Web server) could be eliminated as they are not associated with a director, others (e.g., Apache

Table 1.1 Some examples of ambiguous question and answer pairs. The exact entities mentioned in the pairs are difficult to identify without some additional information on the entities.

Question	Answer
How tall is Michael Jordan?	1.9812m
Who is the director of Apache?	John Lowe
Where is Victoria park located?	London

Cooperation, Apache Software Foundation, and Apache Film) are associated with a director. In addition, John Lowe, which is the answer, may also refer to a darts player, a senator, or a business magnate. The same applies to the third example as there are two famous Victoria parks, and two cities named London in two different countries (in Canada and England).

The answer extraction process by the QA system deals with an internal reasoning scheme to find the answer. Although information from the reasoning scheme could be used as additional information, it is not user friendly when presented as additional information with the answer as it might be overly technical, hence out of context and in some cases it may even confuse the user. In this case, if we provide the descriptions of the entities mentioned in the question in a natural language, it aids the end-users to validate the information as well as grasp new knowledge related to the question.

Even though providing descriptions of the entities contained in the question is promising, the accessibility of the information remains a challenge. Previously, the information was accessed via mining unstructured text for entity descriptions, however, this tends to introduce outliers and noise, hence has been largely ineffective. In the recent times, QA systems have started a paradigm shift towards the use of Linked Data as the source of information. This information source enables us to select structured units of information instead of sifting through a collection of unstructured text. This has made both searching and manipulation of information much more efficient and accurate,

and consequently this has propelled a massive growth of Linked Data resources due to an increasing number of applications using Linked Data information sources.

This thesis investigates and presents a framework, named RealText, which formulates an informative answer as natural language sentences, emanating the characteristics of a human generated answer to a question. We use the factoid questions (questions which require single fact answers) from the Question Answering over Linked Data (QALD) question set (Unger et al., 2012) and their answers as the dataset to extract relevant information from DBpedia in order to construct information rich answers to the question instead of the existing *factoid*, or single fact answers. To do this, we used the structure of the question, the information contained in the question (i.e., entities mentioned in the question), and the pre-existing answer to the question as the basis for selecting information as *triples* in DBpedia. The triples are represented as subject, predicate and object, and has become a de facto format for knowledge representation since it can be used to encode fact units and categorised into conceptual hierarchies defined in an *ontology*. In addition, a knowledgebase encoded as linked triples, is queryable using very basic joins via a SPARQL Protocol and RDF Query Language (SPARQL) query. This makes a linked triple, referred to as *Linked Data*, an attractive natural language resource. In this research, we principally focus on techniques to convert the triples into natural language sentences, rather than a selection of the appropriate triples.

We introduce information rich answers as *Informative Answers* as defined below.

An Informative Answer is an answer in the form of natural language sentences which include the answer and additional information on the entities mentioned in the question and the answer, in a format akin to humans.

According to the above definition, an informative answer is composed of two distinct parts. The first part is the natural language sentence that embeds the factoid answer which we define as the *answer sentence*. The answer sentence further naturalizes the

factoid answer by presenting it as a sentence, specifically generated for the question being processed. The second part contains the natural language paragraphs which provide additional relevant information on the entities mentioned in the question and the answer. The information is based on the DBpedia triples which are transformed into natural language using Natural Language Generation (NLG) techniques. Natural Language Generation is the process of interconnected tasks with the objective of transforming a structured representation into a natural language. The process includes the following tasks to generate an informative answer.

- Generating a sentence which embeds the answer using the linguistic structure of the source question (Answer Sentence Generation)
- Transforming triples into natural language sentences (Lexicalization)
- Organizing sentences by merging and splitting them into paragraphs (Aggregation)
- Generating referring expressions to previously mentioned entities (Referring Expression Generation)
- Organizing and presenting the final answer in different document formats (Structure Realization)

Table 1.2 shows an example output that can be expected from such a framework. The question, SPARQL query, and the factoid answer are available to the framework resources, since it starts from the answer presentation stage, which is the last step of a QA system. In the example given, the framework first produces the answer sentence, which is a direct natural language sentence containing the answer. The framework then extracts the two entities, Berlin and Klaus Wowereit, and the triples which belong to these entities are retrieved from DBpedia. Figure 1.1 shows a portion of triples which

Table 1.2 An example output that was generated from our framework for a test question. The framework uses the question, factoid answer, and the SPARQL of the question to extract information which were then converted to natural language to generate the informative answer.

Question	Who is the mayor of Berlin?
SPARQL of the question	<pre> PREFIX dbo: <http://dbpedia.org/ontology/> PREFIX res: <http://dbpedia.org/resource/> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> SELECT DISTINCT ?uri ?string WHERE { res:Berlin dbo:leader ?uri . OPTIONAL { ?uri rdfs:label ?string. FILTER (lang(?string) = 'en')} } </pre>
Factoid Answer	Klaus Wowereit (http://www.dbpedia.org/resource/Klaus_Wowereit)
Informative Answer	<p>The mayor of Berlin is Klaus Wowereit.</p> <p>Berlin is an administrative region. It is a state in Germany. Its area total is 891.85 million m². Its elevation is 34.0 m. It is governed by Klaus Wowereit. Its population total is 3.539 million. Its area code, and postal code are respectively 030, and 10001-14199.</p> <p>Klaus Wowereit is an office holder. He was born on September 30, 1953 in West Germany. His nationality is Germany. He attended Free University of Berlin. He is a member of Social Democratic Party of Germany. Klaus was the Member of the Berlin House of Representatives, the Vice Chairman of SPD, and the Governing Mayor of Berlin. His term period is October 26, 2011 to December 11, 2014.</p>

belong to these two entities as reported in DBpedia. The retrieved triples are then transformed into natural language descriptions using the processes described earlier.

The framework generated informative answers (answer containing both answer sentence and entity descriptions) for 41 questions out of the 52 question dataset. The answers were evaluated using human evaluation with 14 participants under three criteria; readability, accuracy, and informativeness on a five (1: lowest, 5: highest) point scale.

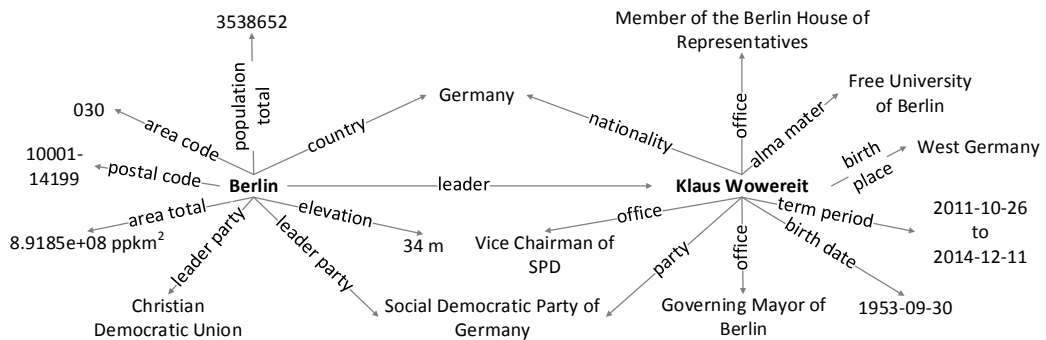


Fig. 1.1 An example scenario depicting the triples associated with two linked entities *Berlin* and *Klaus Wowereit*. The ontology class hierarchies of *Berlin* and *Klaus Wowereit* are, *Place*→*Populated-Place*→*Region*→*Administrative-Region* and *Agent*→*Person*→*Office-Holder* respectively.

The evaluation showed that 95% of the generated answers achieved an average rating of above three for all of the criteria. Furthermore, 39.02% of the generated answers achieved an average rating of above four for readability criteria, while the value for accuracy and informativeness were in the vicinity of 66%.

The rest of the thesis is structured as follows. Chapter 2 presents a review of the literature related to the answer presentation in QA systems. Chapter 3 presents the RealText framework in detail and discusses the theoretical approaches associated with individual modules. Chapter 4 discusses the evaluation of the framework with a focus on evaluation using human participants. We conclude the thesis in Chapter 5 with a summary of the research and overview of the future work.

1.1 Publications from this Research

The research carried out to support this thesis resulted in various publications¹ in refereed journals and conference proceedings. Section 1.1.1 and Section 1.1.2 show the journal publications and conference proceedings produced from this research re-

¹A complete list of publications and technical reports related to the projects carried out during the time of PhD are available at: <http://www.rivinduperera.com/publications/>

spectively. In addition to the research described in this thesis, the tools and techniques developed in this research were also used in various other research projects. Publications related to these projects are listed in Section 1.1.3.

1.1.1 Journal Publications

1. **Perera, R.**, Nand, P., Seki, K. & Burget, R. (Under review) “Please sir, I want some more”: The Art of Asking More Information in Question Answering over Linked Data.
2. **Perera, R.**, Nand, P. & Burget, R. (Under review) Semantic Web Today: From Oil Rigs to Panama Papers. pp. 1-28.
3. **Perera, R.**, Nand, P., Toshioka, K., Yang, W. & Burget, R., (Under review) RealText Approach towards a Lexicalized DBpedia. pp. 1-22.
4. **Perera, R.**, Nand, P., Seki, K. & Stock, K., (Under review) Answer Presentation in Question Answering Systems: A Survey and Classification of the Literature. pp. 1-37.
5. **Perera, R.** & Nand, P., (2017) Recent Advances in Natural Language Generation: A Survey and Classification of the Empirical Literature. *Computing and Informatics*. **36** (1). pp. 1-32.
6. **Perera, R.**, Nand, P. & Naeem, A., (2017) Utilizing Typed Dependency Subtree Patterns for Answer Sentence Generation in Question Answering Systems. *Progress in Artificial Intelligence*. **6** (2). pp. 105-119.
7. **Perera, R.**, Nand, P. & Klette, G., (2016) RealText-lex: A Lexicalization Framework for RDF Triples. *Prague Bulletin of Mathematical Linguistics*. **106** (1). pp. 45-68.

1.1.2 Conference Proceedings

1. **Perera, R.** & Nand, P., (2017). An Ensemble Architecture for Linked Data Lexicalization. In: *18th International Conference on Computational Linguistics and*

Intelligent Text Processing (CICLing). Budapest, Hungary. Springer International Publishing.

2. **Perera, R.** & Nand, P., (2016). Lexicalizing Linked Data towards a Human Friendly Web of Data. In: *28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. San Jose, USA. IEEE Press.
3. **Perera, R.** & Nand, P., (2016). Generating Answer Sentences in Question Answering Systems. In: *28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. San Jose, USA. IEEE Press.
4. **Perera, R.** & Nand, P., (2016). Answer Presentation in Question Answering over Linked Data using Typed Dependency Subtree Patterns. In: *Open Knowledge Base and Question Answering Workshop collocated with 26th International Conference on Computational Linguistics (COLING)*. Osaka, Japan. Association for Computational Linguistics (ACL).
5. **Perera, R.**, Nand, P. & Klette, G., (2016). Lexicalizing DBpedia with Realization Enabled Ensemble Architecture: RealText_{lex2} Approach. In: *15th International Semantic Web Conference (ISWC)*. System Demonstration. Kobe, Japan.
(Travel award funded by Semantic Web Science Association (SWSA) & National Science Foundation (NSF))
6. **Perera, R.**, Nand, P. & Klette, G., (2016). Enriching Answers in Question Answering Systems using Linked Data. In: *15th International Semantic Web Conference (ISWC)*. System Demonstration. Kobe, Japan.
(Travel award funded by Semantic Web Science Association (SWSA) & National Science Foundation (NSF))
7. Nand, P. & **Perera, R.**, (2016). MineYourText: Bridging the Gap between Natural Language and Linked Data. In: *Linked Startup Workshop collocated with 15th International Semantic Web Conference (ISWC)*. Kobe, Japan.
8. **Perera, R.** & Nand, P., (2015). Answer Presentation with Contextual Information: A Case Study using Syntactic and Semantic Models. In: *28th Australasian Joint Conference on Artificial Intelligence (AI)*. Vol. 9457. Canberra, Australia. 30 November–4 December 2015. Heidelberg: Springer International Publishing. pp. 476-483.

9. **Perera, R.**, Nand, P. & Klette, G., (2015) RealText_{lex}: A Lexicalization Framework for Linked Open Data. In: *14th International Semantic Web Conference (ISWC)*. System Demonstration. Bethlehem, Pennsylvania, USA.
10. **Perera, R.** & Nand, P., (2015). A Multi-Strategy Approach for Lexicalizing Linked Open Data. In: *16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*. Vol. 9042. Cairo, Egypt. Heidelberg: Springer International Publishing. pp. 348-363.
11. **Perera, R.** & Nand, P., (2015) RealText_{asg}: A Model to Present Answers Utilizing the Linguistic Structure of Source Question. In: *29th Pacific Asia Conference on Language, Information and Computation (PACLIC)*. Shanghai, China. 30 October–02 November 2015. Association for Computational Linguistics (ACL). pp. 206-214.
12. **Perera, R.** & Nand, P., (2015) Selecting Contextual Peripheral Information for Answer Presentation: The Need for Pragmatic Models. In: *29th Pacific Asia Conference on Language, Information and Computation (PACLIC)*. Shanghai, China. 30 October–02 November 2015. Association for Computational Linguistics (ACL). pp. 197-205.
13. **Perera, R.** & Nand, P., (2015) Generating Lexicalization Patterns for Linked Open Data. In: *Second Workshop on Natural Language Processing and Linked Open Data collocated with 10th Recent Advances in Natural Language Processing (RANLP)*. Hissar, Bulgaria. 5-11 September 2015. Association for Computational Linguistics (ACL). pp. 2-5.
14. **Perera, R.** & Nand, P., (2014). RealText_{cs}- Corpus based domain independent Content Selection model. In: *26th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. Limassol, Cyprus. 10-12 November 2014. IEEE Press.
15. **Perera, R.** & Nand, P., (2014). The Role of Linked Data in Content Selection. In: *13th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. Gold Coast, Australia. 1-5 December 2014. Springer International Publishing.
16. **Perera, R.** & Nand, P., (2014). Interaction History Based Answer Formulation for Question Answering. In: *5th International Conference on Knowledge Engineering and Semantic Web (KESW)*. Kazan, Russia. 29 September 2014. Springer International Publishing, pp. 128 - 139.

1.1.3 Other Related Publications

1. **Perera, R.** & Nand, P., (2015). KiwiLOD: Transforming New Zealand Open Data to Linked Open Data. Auckland University of Technology, New Zealand. (AUT summer research award)
2. Nand, P. & **Perera, R.**, (2015). An Evaluation of POS Tagging for Tweets Using HMM Modelling. In: *38th Australasian Computer Science Conference (ACSC)*. Sydney, Australia. 27-30 January 2015. Australian Computer Society, pp. 83 - 89.
3. Nand, P., **Perera, R.** & Klette, G., (2015). A Tweet Classification Model Based on Dynamic and Static Component Topic Vectors. In: *28th Australasian Joint Conference on Artificial Intelligence (AI)*. Canberra, Australia. 20 November - 04 December 2015. Springer International Publishing, pp. 476 - 483.
4. Nand, P., **Perera, R.** & Lingmin, H., (2014). A Multi-Strategy Approach for Location Mining in Tweets: AUT NLP Group Entry for ALTA-2014 Shared Task. In: *12th Australasian Language Technology Workshop (ALTA)*. Melbourne, Australia. 26-28 November 2014. Association for Computational Linguistics (ACL).
5. Nand, P., **Perera, R.** & Lal, R., (2014). A HMM POS Tagger for Micro-blogging Type Texts. In: *13th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. Gold Coast, Australia. 1-5 December 2014. Springer International Publishing, pp. 573 - 586.

Chapter 2

Literature Review

This chapter explores various answer presentation models which are related to the work being introduced in this thesis. The models discussed are organized as a taxonomy of works which provides a pathway describing how different strategies evolved over time. The taxonomy comprised of three main themes and sub-classifications, as shown in Fig. 2.1 in a tree view.

The rest of the chapter is organized as follows. Section 2.1 provides explanations for the different terms that are used throughout this review. Section 2.2, Section 2.3 and Section 2.4, discuss the three main themes of answer presentation by analysing a number of different systems. Section 2.5 discusses the evaluation metrics employed in

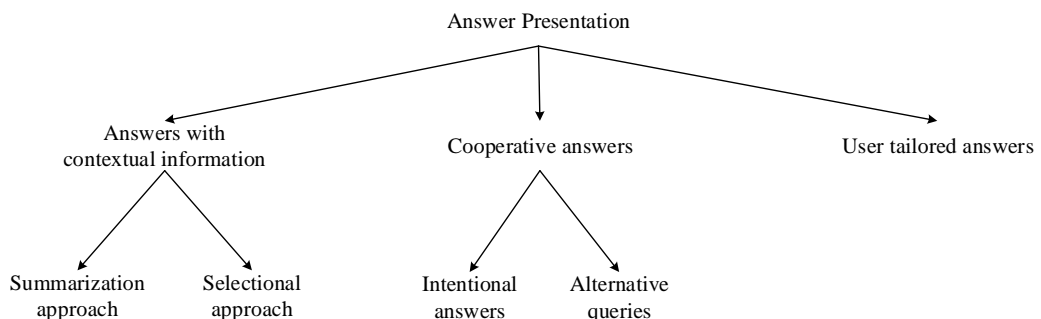
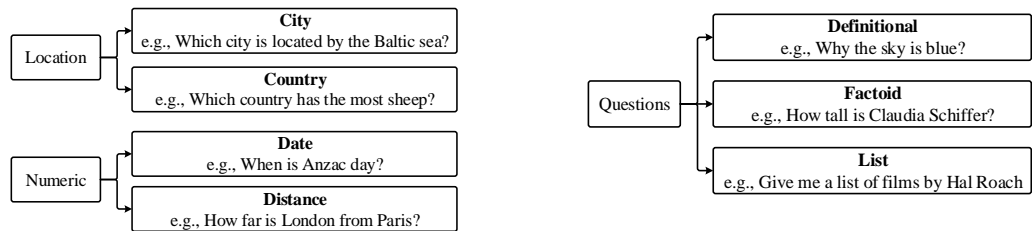


Fig. 2.1 Taxonomy of answer presentation research



(a) *Semantic class based* classification for two sample semantic classes; location and numeric

(b) *Information need based* classification. The factoid, list, and definitional are the three main types of classes in information need based classification.

Fig. 2.2 Question classification based on, *semantic classes* (left) and *information need* (right)

answer presentation. We explore the widely used human evaluation as well as automatic metrics that can be utilized in answer presentation. Section 2.6 concludes the chapter with a summary of findings.

2.1 Definitions

Question Answering (QA) is an area of research which aims to automate the provision of answers to questions posed in a natural language.

Question classification can be accomplished using two main paradigms. The first paradigm classifies questions into semantic classes imposing constraints on possible answers (Li and Roth, 2002). The second paradigm categorizes questions based on the information needed to answer the question. The possibilities for this are whether the question requires a single piece of information (factoid answer), a list of factoid answers, or a definition. Figure 2.2a and Fig. 2.2b depict the two main classification schemes with examples of sub-classification for *semantic class* and *information need based* classification.

Factoid question is a question which requires a single piece of information as the answer. Factoid questions may require properties of an already mentioned entity (e.g., “*how tall is Natalie Portman?*” \Rightarrow *1.6m*) or can expect a new entity as the answer (e.g., “*who is the founder of Amazon Inc.?*” \Rightarrow *Jeff Bezos*).

Answer presentation is a sub-area in QA which investigates the techniques to present answers beyond just raw answers. The initial steps in QA focus on extracting the answer(s), while the later answer presentation stage focuses on presenting an enriched human friendly answer. Many researchers (e.g., Maybury (2008); Mendes and Coheur (2013)) support the use of an answer presentation module for “humanizing” the output from a QA system.

Information source is a collection of information from which the QA system can extract an answer to a given question (Kolomiyets and Moens, 2011). In addition to this main usage, the same information source can be used at the answer presentation stage, to retrieve additional contextual information in order to enrich the answer. Section 2.2 discusses the existing models in answer presentation with contextual information. Recent trends in QA have resulted in a diverse set of information sources; among them is the trend to utilize the massive Linked Open Data cloud as an information source which is described in Chapter 1.

2.2 Answers with Contextual Information

One of the fascinating abilities in the human QA process is to present contextually related information alongside the answer to a question. For example, a person asking the question “*who is the founder of Apple Inc.?*” is seeking “*Steve Jobs*” as the answer. However, assume that the person answering this question has a reasonable amount of background information on the entities mentioned in the question and answer (“*Apple*

Inc., Steve Jobs”). In such a scenario, he/she may answer this question with the correct answer, but also provide a set of related and relevant facts (e.g., basic facts about the “*Apple Inc.*” and “*Steve Jobs*”).

The need for providing contextual information in answer presentation was first brought into discussion by Lin et al. (2003). Their investigation focused on whether users would prefer answers with additional information acquired from the same source where the answer appears. They conducted a user study where participants were asked to report the preference for four different presentation approaches: exact answer, answer with the source sentence, answer with the source paragraph, and answer with the source document. The results from this study showed that the users preferred answers with the source paragraph the most, and exact factoid answers the least. The study reported that there was a strong preference for answers from QA systems to contain contextual information. The study further revealed that there was less preference for answers in which non-contextual information interleaved with contextual information. This is exemplified by Lin et al. through the result that the documents which inevitably contain non-contextual information are less preferred compared to the paragraphs.

The study by Lin et al. is influential in guiding the QA systems to present contextual information. However, contextual information presentation can bring two new challenges to the QA domain. Firstly, if the information source is unstructured text, the development of systems to summarize the text to retrieve contextual information is needed. However, summarization approaches are unaware of the intent of the question so that the relevance of information can be measured against the intent of the question. Intent identification for QA is held as an elusive goal due to the difficulty in extracting limited domain knowledge from a question due to the high degree of variations embedded in syntactic, semantic, and pragmatic aspects of the intent. This challenge can be overcome with a “ranked selectional approach” implemented on a

structured information source (e.g., select relevant triples from a Linked Data resource for a particular question/answer. See Section 2.2.2 for further details). In this approach we first extract entities mentioned in the question and the corresponding answer, and then query the structured information source to extract the information on the entities (e.g., query a Linked Data resource to extract triples with subject “*Steve Jobs*”). This new approach raises the additional need to lexicalize the structured information in order to be able to present it as natural text segments. This is an additional challenge that is not present in approaches which extract existing sentences to be presented as contextual information.

The following sections focus on each of these approaches in turn. We first discuss existing models which use the summarization approach and then move to the selectional approach.

2.2.1 Summarization Approach

Document summarization (automatic summarization) focuses on distilling the most important information from an unstructured text resource (Jurafsky and Martin, 2000). Both answer extraction and presentation use aspects of summarization as part of the process. In this section we discuss only the models that utilize summarization as a technique for answer presentation with contextual information.

Bosma (2005) introduced the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) based summarization technique to present answers. RST is an established technique in discourse processing which establishes relations between text segments within the discourse. The segments are referred to as the nucleus and satellite. The nucleus is the text segment that is central to the purpose of the writer, and the satellite helps to increase the understanding of the message delivered in the nucleus (Jurafsky and Martin, 2000). The relation (rhetorical relation) that gets established between

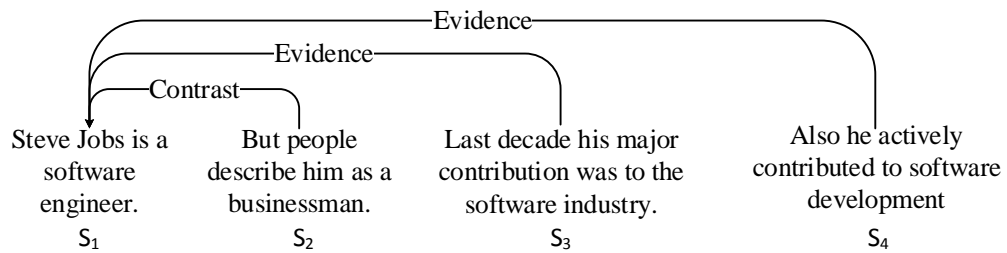


Fig. 2.3 An example of an annotated text segment with rhetorical relations

the nucleus and the satellite can be one of the 23 types defined by RST (Mann and Thompson, 1988). Figure 2.3 shows an example of an RST annotated paragraph of text, which must be created manually. The figure depicts the existence of two types of rhetorical relations in a passage composed of four sentences.

Bosma (2005) uses an RST annotated document collection and then builds a weighted graph using the Dijkstra shortest path algorithm. The weighted graph is used to select the sentence which contains the answer and this is used as the starting point of the answer. For instance consider the example scenario depicted in Fig. 2.3. If we consider the question, “*what was Steve Jobs main contribution in the last decade?*”, then the answer sentence will be S₃. Based on the sentence S₃, weights can be allocated to the graph. The candidate sentences (sentences with higher weights) will be selected for the summarization using these weights. An application of Bosma’s (2005) answer presentation model is used in the IMIX project (Theune et al., 2007). In this project, the approach is directly used to present an answer to the user with the discourse structure of the document.

Bosma (2005) specifies that an RST relation annotated document is a prerequisite for the model and does not consider the RST annotation as an error prone extra effort. RST annotation is a tedious task and can only be accomplished using experienced linguists (Carlson et al., 2003). Due to this, RST based summarization cannot be fully automated

without human intervention. The challenge to automate the complete summarization process is a major drawback in Bosma's (2005) model for answer presentation.

A strategy to overcome the human intervention reported in Bosma's (2005) model is to integrate statistical Natural Language Processing (NLP) models in the summarization task. Demner-Fushman and Lin (2005; 2006) present an ensemble of classifiers which are trained to extract salient sentences from MEDLINE¹ citations to build a summary for a given biomedical question. This ensemble approach is comprised of six different classifiers; a rule based classifier, a unigram classifier, an n-gram classifier, a position classifier, a document length classifier, and a semantic classifier. Both unigram and n-gram classifiers are based on Naïve Bayes, however, they work with different feature sets. The rule based classifier is developed utilizing a set of rules defined by an experienced nurse. The position classifier uses a maximum likelihood estimate of a sentence being a salient sentence based on its position in the MEDLINE abstract. The document classifier returns the smoothed probability (using add one smoothing) that a document with a particular length (measured by number of sentences) contains the salient sentence. The semantic classifier outputs a score based on the presence of Unified Medical Language System (UMLS) concepts. Except for the rule based classifier, the remaining five classifiers are based on machine learning and statistical NLP methods. Demner-Fushman and Lin evaluate the model using the PubMed abstracts as a baseline. The results shows that the proposed approach outperforms the PubMed abstract based answer presentation. However, Demner-Fushman and Lin (2005) fail to provide details on the individual performance of the classifiers. Another notable drawback in this study is the absence of answer quality evaluation using readability and informativeness criteria. Since the summarization is targeting a more human-friendly presentation of answers, the need for an evaluation based on linguistic qualities is essential.

¹<http://www.nlm.nih.gov/pubs/factsheets/medline.html>

MedQA (Yu et al., 2007) is another QA system with summarization based answer presentation which studies the user preference of answers through a qualitative evaluation. MedQA applies centroid-based summarization to select the most salient sentences and then generates the summary aggregating these sentences. To facilitate the centroid based summarization, a set of clustered sentences are required. MedQA clusters sentences using two hierarchical clustering algorithms; group-wise average and single-pass clustering. The group-wise average considers the entire set of sentences and depends on the Term Frequency - Inverse Document Frequency (TFIDF) to measure the similarity between two sentences to decide whether they are candidates to merge and form a cluster. The TFIDF is then recursively computed within clusters to measure whether there are any candidate clusters to be merged. Single-pass clustering is initiated with one randomly selected sentence and forms clusters incrementally by computing the similarity score between a new sentence and the selected ones. Of these two approaches, the group-wise average is more computationally heavy ($\frac{1}{2} \times N \times (N - 1) + N^3$, for N sentences) than the single-pass clustering ($\frac{1}{2} \times N \times (N - 1)$). However, the group-wise average outperforms single-pass clustering (Hatzivassiloglou et al., 2000). Therefore, MedQA applies two approaches selectively based on the situation. Specifically, MedQA applies single-pass clustering when a large number of sentences are present and applies group-wise average for smaller numbers of sentences. This selective summarization attempts to balance accuracy and performance by switching between the algorithms. Although such an approach can achieve a certain level of accuracy, the system cannot be consistent in the nature of summarization due to the switching mechanism.

Apart from its methodological issues, MedQA introduces the importance of human evaluation to determine the answer quality. The evaluation dataset contains 12 questions and is evaluated using a five point Likert scale (“1” represents poorest and “5” represents

Table 2.1 Evaluation results of the MedQA adapted from Yu et al. (2007). Time spent (in seconds), quality of answer, and ease of use are shown. Quality of answer and ease of use are measured on a five point Likert scale (“1” represents the poorest and “5” represents the best)

	Google	MedQA	OneLook	PubMed
Time spent (seconds)	69.6	59.1	83.1	182.2
Quality of answer	4.90	2.92	2.77	2.92
Ease of use	4.75	4.0	3.9	2.36

the best). Table 2.1 shows the evaluation result of MedQA compared to Google², OneLook³, and PubMed⁴. However, this evaluation suffers from two major weaknesses. Firstly, the inter-annotator agreement is not computed among the human evaluators. Although the participants were carefully selected, the absence of this measure makes it difficult for its reliability to be validated. Secondly, the evaluation utilizes a relatively small test set (12 questions) which cannot be successfully used to interpret results generalizing to the entire domain.

Cao et al. (2011) introduce an improved presentation strategy as well as a novel summarization mechanism in the AskHERMES system. The presentation approach in the AskHERMES system is initiated by identifying content-rich keywords extracted from the question. Each answer retrieved for these content terms is categorized under that content term. The phenomena brought into attention by Cao et al. is that this approach will help users to identify preferred answers easily as they can consult the content-rich terms as an index. The topical clustering of the answers based on the content terms is carried out using hierarchical clustering with the extracted terms as the labels. However, when transforming the extracted terms into labels, the synonyms of different terms are also considered. In addition, AskHERMES also carries out further clustering within the main clusters if the number of passages for a cluster exceeds a

²<http://google.com/>

³<http://www.onelook.com/>

⁴<https://www.ncbi.nlm.nih.gov/pubmed>

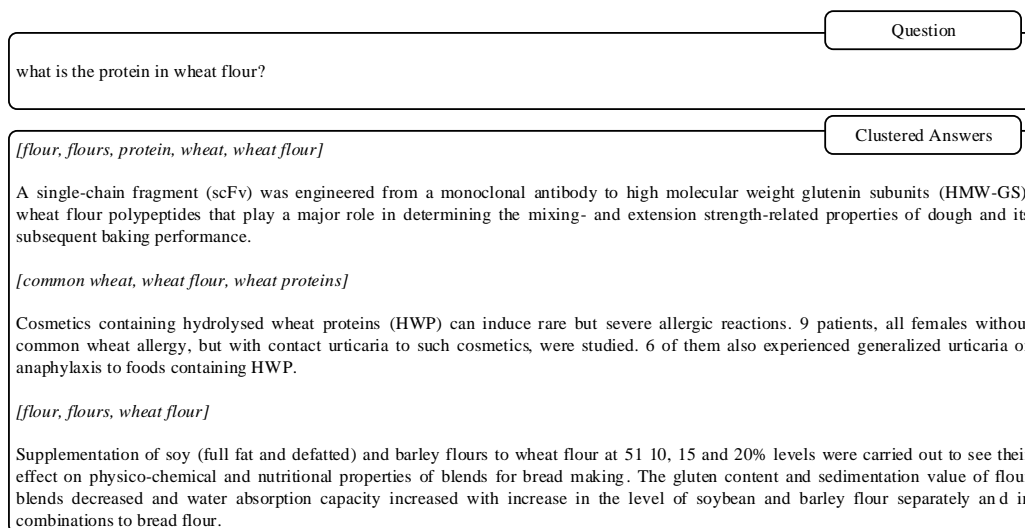


Fig. 2.4 The result for the question “*what is the protein in wheat flour?*”. The result is obtained from the AskHERMES demo hosted in <http://www.askhermes.org/index2.html>

particular threshold which is empirically assigned to five. This type of clustering can help the users to easily navigate through the answers and find the most relevant ones.

Table 2.2 describes the summarization based answer presentation approaches with the details on the approach used, evaluation, and the limitations. Although the summarization approach can present additional information to the user, the relatedness of the summarized information to the question/answer is not always guaranteed. This is because the summarization is not working at the information unit level and therefore is unable to extract information from a sentence which carries both relevant and irrelevant information. The framework being proposed in this thesis addresses this challenge by working at the information unit level which are related to the answer and/or the question. The Linked Data resource being utilized in the framework contains the semantics and metadata embedded into the information units, which is extremely important for finalizing the answer presentation.

Table 2.2 Summary of the answer presentation approaches using summarizing the contextual information

Citation	Approach	Evaluation	Weaknesses
Bosma (2005)	Uses Rhetorical Structure Theory (RST) to identify links between a nucleus and satellite. Identifies key sentences using a weighted graph withijkstra's shortest path algorithm.	Uses a human evaluation to rate the generated answers using a Likert scale. Bosma mentions users reported that answers contained more useful information and less irrelevant information.	Requires manually annotated RST relations, which is tedious and introduces potential errors.
Demner-Fushman and Lin(2005; 2006)	Uses an ensemble of six classifiers to extract salient sentences from medical abstracts . The classifiers are: rule based (from human expertise); unigram; n-gram; position; document length and semantic.	Evaluates the model using PubMed abstracts as a baseline. Results show that the model outperforms the PubMed abstract based answer presentation.	Evaluation do not consider linguistic qualities including readability and informativeness of the answers.
Yü et al. (2007)	Uses centroid based summarisation to select the most salient sentences. Creates clusters of similar sentences using group-wise and single-pass clustering.	Performs human evaluation with 12 questions by comparison with three other approaches (including Google). Google was better for ease of use and quality of answer, but slower.	Small set of sample questions in evaluation.
Cao et al. (2011)	Identifies content rich words and clusters answers under those keywords and their synonyms.	Uses the human evaluation to measure the ease of use, quality of the answer, and overall performance.The approach presented by Cao et al. (2011) has achieved slightly lower values compared to two other systems, Google and UpToDate.	The clustering based presentation model may not be useful for users who seek summarized answers. Furthermore, clustering terms need to be associated with synonyms and related concepts.

2.2.2 Selectional Approach

The selectional approach is based on choosing individual units of information, which are then presented to the user as a descriptive answer. Instead of condensing the information as in the summarization approach, the selectional approach offers greater freedom to select information to associate with the answer. However, to implement the selectional approach, a rich information source and a method of selection must be readily available.

Incorporating a domain ontology where an answer is present and selecting all related concepts with that answer can be considered as a naive approach to select contextual information in the answer presentation stage. Vargas-Vera and Motta (2004) present the AQUA QA system which utilizes the AKT reference ontology as the information source to perform a simple aggregation of ontology nodes which are related to the answer. An example scenario of the answer presentation approach in AQUA is shown in Fig. 2.5. Since the AKT reference ontology is a domain ontology, the selected information provides contextual information. The naive approach presented by Vargas-Vera and Motta (2004) can perform well in the presence of a highly-specified domain ontology for the question being processed. However, finding such an ontology is a major challenge for open domain QA systems and may also be difficult for closed domain QA systems which operate on some domains that suffer from limited resources.

The previously mentioned method of utilizing a domain ontology can be further extended if a knowledge base is used as the source of information. Katz et al. (2007) implement the START QA system to work with a knowledge base where a specially designed framework, Omnibase (Katz et al., 2002), is added to help with selection of relevant information. Omnibase is implemented using the object-property-value data model which makes it easier to retrieve the answer as well as identify additional information related to the answer. In addition to selecting text from the knowledge base, the START QA system can select and extract graphical representations to support the

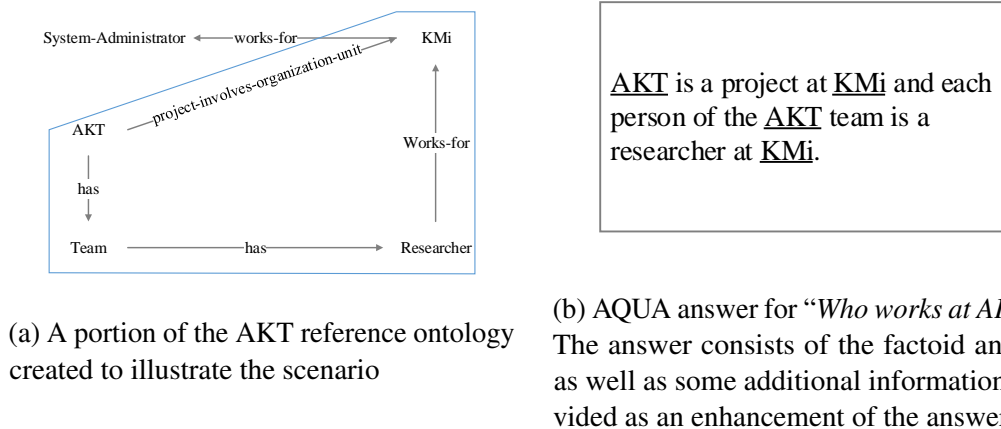


Fig. 2.5 A portion of the AKT reference ontology (ontology nodes related to the answer are outlined) (left) and the answer generated by the AQUA QA system with the output enhancement module (right)

answer. These graphics are extracted using the keywords related to the question and answer. Figure 2.6 shows the output for the question “*What is the capital of USA?*”. According to the figure, the answer generated from the START QA system includes the basic information related to the USA and an image of the USA flag extracted from The World Factbook⁵. The answer is extracted through Omnibase using the START knowledge base. The generated answer has two main drawbacks; firstly there is a lack of additional information about the actual answer (e.g., population of Washington, its location, etc.), and secondly the information provided about the USA is in structured form without lexicalizing it to make it easier for readers to understand.

Duboue and McKeown (2003) adopt a statistical approach to select contextual information to present with the answer. The study focuses on developing a method for acquiring biographical information which is later proposed to integrate to the AQUAINT QA system. The system can then be extended to provide descriptive answers for questions related to biographies. The information source for biography generation is a frame based knowledge base. This is created by transforming text (extracted

⁵<https://www.cia.gov/library/publications/the-world-factbook>

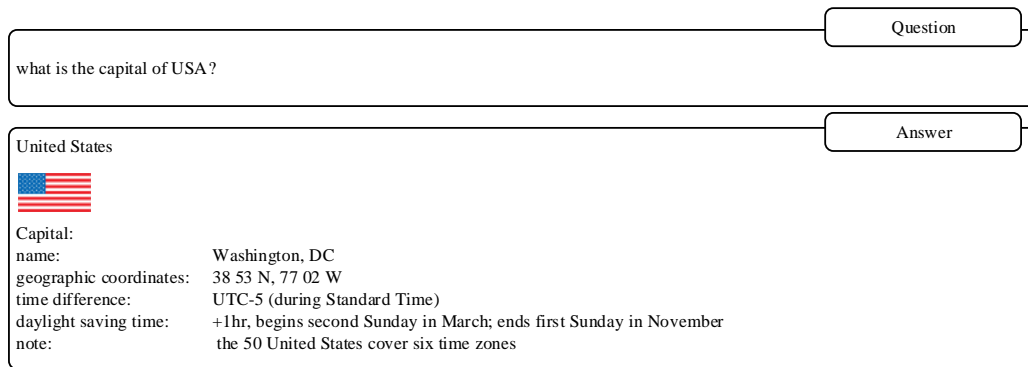


Fig. 2.6 Answer generated from START QA system with information selected from Omnibase

form 1,100 HTML pages containing celebrity fact-sheets) to frames. Duboue and McKeown (2003) attempted to train a model using this information source and target biographies which are collected from *biography.com*. The model proposed by Duboue and McKeown (2003) is composed of two steps to induce a set of rules for selection; in the first step a verbatim based approach is used while in the second step a statistical approach is integrated targeting higher accuracy. The verbatim based rule induction focuses on comparing an input and the respective output to analyse what information is copied as it is. The information that is present in this manner are considered mandatory and a rule is composed to select such information. For example, date of birth and profession may be present in the text as it is, and therefore it is essential to include such information when composing a biography. The statistical selection stage is based on deriving language models from clustered information and using these language models (cross-entropy using bigrams) to analyse correlation between clustering and language models to identify content selection rules.

A key drawback in the approach introduced by Duboue and McKeown is that it does not focus on conditions which determine the generalization of the content selection rules. For example, the biography of a person who has contributed to Information

Technology (IT) should include details of his/her significant contributions. However, if the development dataset does not include such biography, then this approach cannot capture specific, but important details, hence such information will not be included in the generated biography. Therefore, the proposed approach by Duboue and McKeown is largely dependent on the data provided to the system, hence the system is only capable of generating a very general biography. One of the solutions to overcome this challenge would be to consider the classification of people based on an ontology. The existence of such resources can determine information which is important for groups of people. For example, if the system is generating a biography for “Steve Jobs”, then knowing that he is an IT professional would help the algorithm to include and prioritize his contributions to the IT domain. Such a presentation will benefit the users as they can find more specific details in addition to the general information (i.e., birth date, birth place).

Small et al. (2003a; 2003b) present the interactive QA system, HITIQA. HITIQA is designed to answer analytical questions by retrieving additional contextual information. HITIQA initializes the process by retrieving documents using tokens extracted from the question. The retrieved documents are then clustered to identify the information distribution. The clustering is based on n-grams and the base concepts are generated using noun phrases extracted from WordNet (Miller, 1995). However, the information is still not properly segmented to perform a selection with the basic information. This is because the information is embedded in sentences and therefore a sentence may present multiple information units. If a single sentence carries the most important information as well as unwanted information for the question, this would decrease the accuracy of the overall process. On the other hand such sentences cannot be rejected because of the important information they contain. HITIQA addresses this problem by segmenting the sentences into text frames, so that the atomic level selection can be

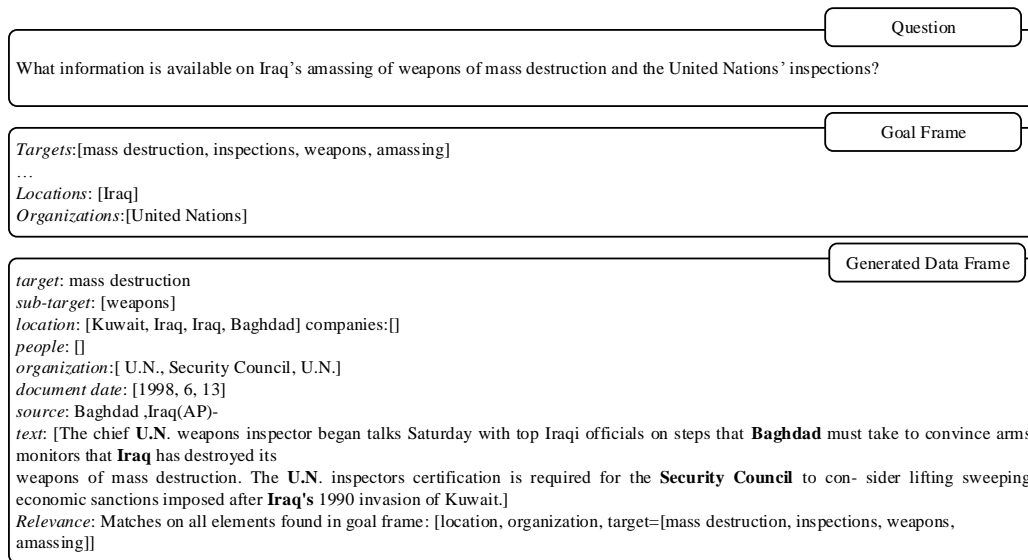


Fig. 2.7 HITIQA data frame generation

performed. The framing is essentially an entity recognition. The main drawback is that entity recognition may not support all of the entities mentioned in the text as it is limited based on its training instances and the templates employed. The frame selection is carried out by scoring relevance through analysing conflicts between a goal frame (created using the question) and the frames created for retrieved text. A sample output expected from the HITIQA is depicted in Fig. 2.7. This shows the goal frame and the generated data frame for the question “what information is available on Iraq’s amassing of weapons of mass destruction and the United Nations’ inspection?”.

Table 2.3 summarizes the answer presentation models discussed in this section. The effectiveness of the information presentation depends on how easily it is disseminated by human end-users. For example, Vargas-Vera and Motta (2004) attempt to naively join ontology concepts to generate an answer with additional information. This is limited to domain ontologies and is not applicable to ontologies whose concepts are not ready to be represented in controlled natural language expressions. Several other limitations associated with current information selection approaches are listed in Table 2.3. In

Table 2.3 Summary of the answer presentation approaches by selecting the contextual information

Citation	Approach	Evaluation	Weaknesses
Vargas-Vera and Motta (2004)	Uses an ontology to find the node that best answers the question, and then generates a chunk of information using related nodes, drawing on the relations and inferences from the ontology specification.	No specific evaluation is reported to find the effectiveness of the answer presentation model.	Requires a very detailed and well specified ontology in order to provide meaningful answers.
Katz et al. (2007)	Uses a knowledgebase to store information in object, property, value form, and matches against a question that is also formulated in the same ternary form, taking linguistic relations like synonymy and homonymy into account.	No specific evaluation is reported to find the effectiveness of the answer presentation model.	The information provided is limited, and in a structured form, rather than natural language, so may be considered more difficult for users to navigate.
Duboue and McKeown (2003)	Populates a frame-based knowledgebase from celebrity fact sheets and attempts to use machine learning to induce a set of rules. Uses biographies from the web to train the system. Rule induction first finds verbatim elements that are included in the final answer and the second uses language models and clustering to find correlated content.	Focuses on a gold set of content selected by human participants. The class-based content selection rules achieve the highest F-Scores of 0.58 and 0.51 in development and test datasets respectively.	The approach focuses only on extracting general rules to present answers when used in a QA system. In addition, an evaluation which focuses on informativeness and linguistic quality can better measure the content instead of a gold standard dataset.
Small et al. (2003a; 2003b)	Retrieves documents using tokens extracted from the question, then clusters to identify information distribution. Creates a goal frame from the question, retrieval frames, and calculates a relevance score.	A human evaluation which resulted in an average score of 5.8 on a seven point scale. No inter-rater agreement is provided.	Framing is based on entity recognition, the success of which may be limited by the availability of training data and templates used.

order to overcome these limitations this thesis explores methods of generating patterns that can transform the information (i.e., Linked Data triples) into natural language with an ensemble of approaches. We also explore the methods of further realizing the generated language, so that the answers can be presented to the user in a form which depicts the tone and tenor of human generated text. Further operations, such as aggregation, referring expressions, and structure realization, are investigated to further improve the presented answer to make it more akin to a human answer.

2.3 Cooperative Answers

QA systems are generally built to provide an answer only if it exists. However, cooperative QA systems collaborate with the user and provide a useful message if no answer can be found or provide other options that are associated with the answer which can be useful to the user. Furthermore, cooperative QA systems can provide additional information to the user based on the collaboration that they managed during the QA process with the user. Cooperative QA systems can provide: intensional answers, alternative queries, or qualified answers (Benamara, 2004; Melo et al., 2013). In this review we consider cooperative QA systems that provide the first two of the above types which directly support answer presentation. Qualified answering is a model used to select answers based on a number of constraints defined by the end user (Gaasterland and Lobo, 1994), and hence does not put emphasis on the presentation component in a QA system. However, qualified answering can be utilized as an underlying function to support user tailored answering which is another answer presentation model which is discussed in detail in Section 2.4. A cooperative answer is defined as an indirect answer and may contain failure information and other possible directions if the QA system fails to answer the question. Additionally, cooperative answers can carry additional information which is unsolicited. Such a presentation of information can advance the

Table 2.4 Extensional and intensional answers for the question “which countries use the Euro?”

Question	Which countries use the Euro?
Extensional answer	Austria, Belgium, Cyprus, Estonia, Finland, France, Germany, Greece, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Portugal, Slovakia, Slovenia, and Spain.
Intensional answer	All Eurozone countries use the Euro

knowledge of the user as well as help him/her to validate the answer based on common background knowledge.

2.3.1 Intensional Answers

In general, a QA system is designed to provide an answer set which satisfies a given query. These answers are called extensional answers. However, extensional answers lack the characteristics and common features of answer sets provided by an intensional QA system. Intensional answers cover the missing information and help the user to understand the question well. For example, the extensional answer for the question provided in Table 2.4 is a list of 19 countries. In comparison, the intensional answer goes beyond the list and provides the common feature (e.g., Eurozone countries \xrightarrow{use} The Euro) of the answer set.

An intensional answer can move beyond simple data generalization to statistical summarization and generalized rule extraction which can ultimately provide more useful information than the lengthy enumerations given in extensional answers (Benamara, 2004). The main advantage of intensional answers is that they support end users to resolve misconceptions. In addition, they may help the users to understand the structure of the knowledge base being used in the system. On the other hand, the quality and quantity of the underlying knowledge base essentially affects the quality of the

intensional answer. This is because the intensional answer is generated by applying various levels of generalization to the knowledge base.

Most researchers investigating intensional QA systems have utilized database systems for implementation. Although there are some differences between QA and database querying, they share the same core concept of retrieving and presenting information. Essentially, a database contains schemas which define integrity constraints, relationships, and data classes. According to the aforementioned definition of intensional and extensional answers, schema definitions constitute the intensional information while the data in the database represents the extensional answers for a particular database query. Therefore, a database query is essentially an intensional statement that requires extensional information from the database (Motro, 1994).

Motro (1994) compares the different approaches that have been developed to enhance database query results with additional intensional statements. He classifies approaches to produce intensional answers into four main categories:

- Data model with intensional information
- Including extensional answers in intensional answers
- completeness of intensional characterization
- independence from extensional information

There is a wide variety of research attempts which adhere to the above four aspects. However, the core idea of the process is to enhance the answer with the application of some level of generalization. For example, assume that we execute the query, “*who earns over \$2000?*” in a database system which contains employee records. The intensional answer can take the advantage of other information present in the specified data table or other tables that have relations with the specified table. In which case the intensional answer of form “*All software engineers except John White who is currently*

in his probationary period get salary over \$2000. The average salary is \$2300” can be expected which generalizes the results as well as does an additional step of providing simple statistics related to the query.

A comprehensive systematic study of building intensional answers in natural language QA systems was reported by Benamara (2004) through the implementation of the WEBCOOP system. In this approach, Benamara (2004) derives intensional answers from the extensional answers provided for the question using intensional calculus. This study also provides an important perspective on intensional answers by classifying them into five main categories which are identified by analysing a corpus of frequently asked questions. Table 2.5 depicts the intensional answer categories identified by this study. It is clear that the majority of intensional answers are based on some kind of generalization applied on the extensional answers provided for a particular question.

WEBCOOP employs two main modules in the intensional answer generation process. In the first module the system determines the content that needs to be present in the intensional answer. The second module is a template based approach that is used to generate the intensional answer. The content determination starts with a given set of extensional answers and search for the element that is suitable to generalize. The main drawback in this step is that in order to determine a suitable content, there must be a number of extensional answers. Although Benamara (2004) predefines this value to a list of 10 answers in the experiment, most of the factoid answers do not contain this number of answers. Therefore, the whole process fails to generate an intensional answer. In the next step the level of abstraction is determined utilizing the variable depth intensional calculus. Finally, the determined content is presented in natural language using a set of predefined templates.

In another study, Benamara (2002) explains the process of forming a semantic representation for questions and building a semantically rich knowledge base. The main

Table 2.5 Intensional answer categories derived from analysing a corpus of 350 frequently asked questions. The last column shows the representation of each category in the corpus.

Category	Description	Example	Rep. (%)
Introducing higher level concepts	Higher level information is identified based on existing knowledge	<i>Question:</i> Who is Steve Jobs? <i>Answer:</i> Steve Jobs is the founder of Apple Inc. <i>Intensional Answer:</i> Steve Jobs is a founder of a Tech Company	5%
Reorganizing data	Organizing the data to generate meaningful and useful answers	<i>Question:</i> What are the BMW car models? <i>Answer:</i> BMW M3, BMW X1, BMW i3, ... <i>Intensional Answer:</i> Sorted set of BMW cars based on production year/engine capacity	26%
Generalization	Builds summarized answers from the extensional answer set	<i>Question:</i> In which countries is Theravada Buddhism followed? <i>Answer:</i> Sri Lanka, India, Nepal... <i>Intensional Answer:</i> In South-east Asia	43%
Quantification	Generates an answer with quantifiers such as “a few” and “some”	<i>Question:</i> Are all Asian countries densely populated? <i>Answer:</i> India, China. <i>Intensional Answer:</i> Only some Asian countries	21%
Correlation	Generates answer based on relations that appear in question	<i>Question:</i> How many students and teachers will participate in the trip? <i>Answer:</i> About 10 (based on current data) <i>Intensional Answer:</i> It depends on the number of people participating in the team	5%

advantage of this method is that the generalization operation to build intensional answers becomes more accurate and efficient as semantically based inference can be executed on the information contained. Benamara (2002) transforms the questions to represent the question as a conceptual category, semantic type, or context representation. The conceptual category is derived using question classification while the semantic type is determined from the answer. The context representation reformulates the given question into a semantic form using Lexical Conceptual Structures (LCS). The knowledge base contains a domain ontology, facts, rules, and integrity constraints. Given the above formalism, generalizations can be easily acquired by checking higher level semantic classes in the knowledge base ontology for the semantic classes that are mentioned in the question. Although Benamara (2002) employs semantic structures for the cooperative answers, the process of transforming questions to logical structures is an extremely difficult task.

While most of the work on intensional answers focus on textual interpretations, Moriceau (2006) illustrates the methodology of building intensional answers using numerical data for QA systems. In essence, Moriceau explores the methods of integrating answers to form a coherent core from the candidate answers assuming that all the candidates are equally important for the question. Such an answer can also be thought of as another dimension of an explanatory answer. Although Moriceau's work focuses on numerical values, the research also addresses answer integration for textual questions. The following four types of answer integration mechanisms are introduced which have their roots in a previous study by Webber et al. (2002):

Inclusion The inclusion criteria integrates answers if an answer entails another answer.

Moriceau (2006) suggests the integration mapping by considering the ontological relations such as *is-a* or *part-of* relations. For example, if “in London” and “in United Kingdom” are two answers selected by the QA system for the question

“Where is MI5 headquarters?”, then these answers can be linked as London is a part of the United Kingdom.

Equivalence Some candidate answers retrieved by the QA system can also be equivalent in some scenarios. Moriceau (2006) differentiates between two types of equivalence; lexical and inference based equivalence. Lexical equivalence is when two answers are identified as equivalent at the surface level by analysing the two answers. For example, for the question “*who led the continental army?*”, the answers such as “*George Washington, the leader of the continental army*” and “*continental army head George Washington*”, are semantically equivalent. On the other hand some answers may not be equivalent on the surface semantic level, but be detected as equivalence after applying inference. For instance, answers such as “*Apple Inc. is 40 years old*” and “*Apple Inc. was founded on 1976*”, are equivalent for the question “*when was Apple Inc. founded?*”.

Aggregation Some questions may accept multiple answers. These answers can be aggregated through a suitable conjunction operation which provides a more coherent answer to the user.

Alternative This can be used to generate answers based on an inconsistent set of answers available for a question. For example, when asked a question such as “*when is friendship day celebrated in the next 10 years?*”, the user may get answers such as “*6 August 2016*”, “*6 August 2017*”, “*6 August 2018*”, and so on until 2025. However, an alternative integration mechanism would present these answers as a single answer of the form “*Every August 6th for the next 10 years*”.

Answer presentation by integrating candidate answers always requires all answers to be accurate for the given question and in the case where this assumption fails, the final answer becomes invalid or may contain unnecessary information. In QA systems

that work on unstructured text, it generally is not possible to get 100% accuracy in all candidate answers due to the high level of ambiguity that natural language contains. However, Moriceau (2006) focuses on numerical value integration which reduces the aforementioned hurdle to a certain extent. In essence the questions selected for the experiments are questions which focus on numerical answers (e.g., “what is the average temperature in Hawaii?”) . Another factor discussed in this work is the importance of analysing lexical suitability when generating an integrated answer. In particular, Moriceau (2006) proposes the use of verb subclasses according to the variation in lexical items. One such example is the use of verbs such as “*climb*” and “*drop*” with properties like “*gas price*”. Although Moriceau (2006) has not given implementation details on such usage, this requires further work, specifically addressing context in which a lexical item is appropriate, however, this is a challenge in its own right. Previous linguistics work in the use of metaphors in a particular context may be useful for this endeavour (Lakoff and Johnson, 2003).

Cimiano et al. (2008; 2010) report an approach that uses Inductive Logic Programming (ILP) for bottom-up generalization of intensional answers. The approach is implemented as an extension to the ORAKLE QA system which transforms the question into a logical equivalence. The resulting logical form of the natural language question is consumed thereafter by the ILP module to generate an intensional answer. The generalization algorithm is based on the Least General Generalization (LGG). The advantage of LGG is that by nature it looks for the most specific generalization that is true for all extensional answers (Boley and Possner, 1995), which is ultimately what we expect as an intensional answer. Figure 2.8 presents a worked example from the ILP enabled ORAKLE QA system. The figure shows the intensional answer generated for the question “which states have a capital?”, where the intensional answer is presented as a logical statement which expresses that all states have a capital. The research considers

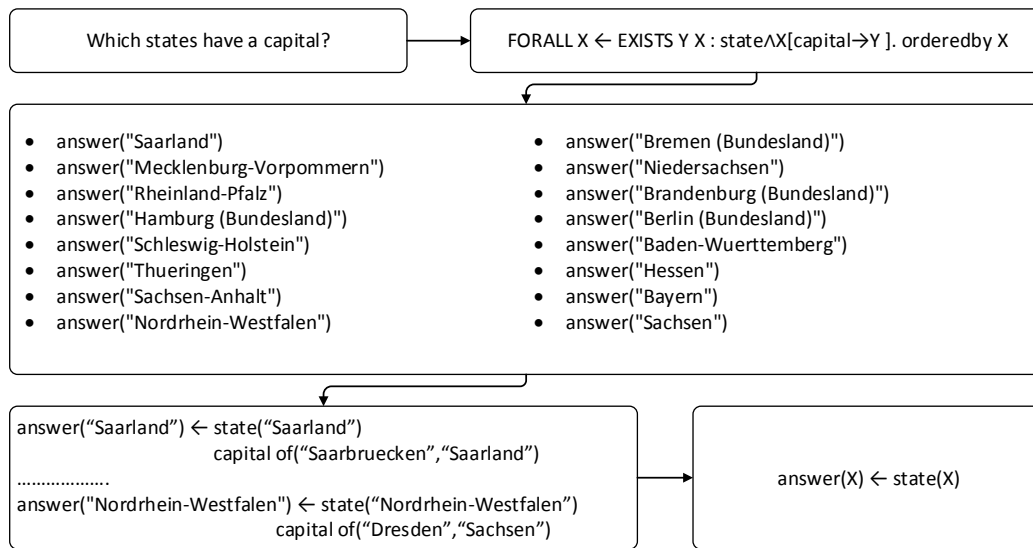


Fig. 2.8 An example intensional answer from the ILP enabled ORAKLE QA system generating a natural language representation of this logical statement as a future work. However, the challenge is that when the intensional answer becomes complex, with logical statements applied on top of each other, the generation procedure will also require specially crafted methods in order to work effectively with such scenarios.

Cimiano et al. (2008) also provide the extracted answer together with the intensional answer generated. These two processes are designed to run in parallel, so that the user does not need to wait until an intensional answer is generated from the system. Furthermore, such a presentation can restrict the approach by foregoing the user expectation in a scenario where a user expects only the factoid answer.

Table 2.6 summarizes the intensional answer based presentation models. Although intensional answers provide valuable information to the user to further enhance his/her knowledge, enhancements are needed in two aspects. On one hand, the power of Linked Data in intensional answer generation is yet to be explored. Linked Data provides an enormous knowledge base with embedded semantics and ontological reasoning which is able to support intentional answer generation, making it felicitous as a source

Table 2.6 Summary of the answer presentation approaches using intensional answers

Citation	Approach	Evaluation	Weaknesses
Benamara (2004)	Finds extensional answers and searches for an element that is suitable to generalise. Then uses intensional calculus to determine the appropriate level of abstraction. The templates are used to present natural language answers.	No specific evaluation is reported to find the effectiveness of the answer presentation model.	Requires a large number (at least 10) of extensional answers, and many questions do not have this many answers.
Benamara (2002)	Transforms questions into a structured form as a conceptual category, semantic type or context representation. Uses a domain ontology which enables the generalisation.	No specific evaluation is reported to find the effectiveness of the answer presentation model.	The process of transforming questions into the semantic structures is very difficult.
Moriceau (2006)	Develops a method to integrate multiple answers into a coherent core, assuming all answers are equal.	No specific evaluation is reported to find the effectiveness of the answer presentation model.	Need to focus on lexicalizing answers. The framework needs experimental evaluation which should be carried out by human participants to analyse the quality of the generated answers.
Cimiano et al. (2008; 2010)	Converts the natural language question into a logical form and uses ILP to generate a generalised logical statement that expresses the intensional answer generated from the extensional answers. Future work will convert the final intensional logical expression into natural language.	The approach was able to generate intensional answers for 169 queries from the 205 query set (82.43% coverage). However, only 140 queries (68.29%) were identified as useful.	Complex logical expressions may require specially crafted methods in order to be converted into natural language. In addition, the model needs a human evaluation which will also analyse agreements between participants.

for natural language answer presentation. In addition, there is a clear necessity to present intentional answers in a natural language form. Benamara (2002) places some importance on presenting the intentional answer in natural language, however, further improvements are needed in order to build a descriptive text which is not covered in Benamara's research. Furthermore, the recent efforts in intentional answer generation by Cimiano et al. (2008; 2010), demonstrate the need to generate intentional answers in natural language. This is discussed as one of the future goals to be attained through further research. The RealText framework proposed in this thesis, concentrates on addressing the two aforementioned issues by firstly utilizing Linked Data (i.e., DBpedia) and secondly transforming the data into natural language to be presented as a descriptive answer. Although, RealText does not focus on intentional answers (instead it generates *informative answers*), the techniques introduced in this framework can be directly utilized to build enhanced intentional answer presentation methods.

2.3.2 Alternative Queries and Query Relaxation

The core objective of alternative querying and query relaxation in cooperative QA systems is to capture information that is related to the answer. However, this should not be confused with the contextual information presentation described in Section 2.2. The aim of the contextual information presentation strategies was to support the retrieved answer with additional information, therefore it is essentially an enrichment of the existing answer.

Alternative querying presents some related answer candidates and information which provide alternative options for the user in the absence of a direct answer or if the direct answer is not satisfactory (Gaasterland, 1997). This is achieved by relaxing the query, so that neighbouring information can be captured and presented to the user. Gaasterland (1997) emphasizes that the availability of semantic information which can

be used to analyse the queries/questions and the search trees is a prerequisite to carry out relaxation. Specifically, if the underlying information source for the querying system is a database, then the database needs to be augmented with a graph of taxonomic relationships between the database relations and attribute values. Once the information source is ready for a relaxation, Gaasterland (1997) proposes three methods to perform the relaxation of the query:

- rewrite the relationship to a general form
- rewrite terms into more general forms
- break the join dependency among the literals in the query

Benamara and Saint-Dizier (2004) present the query relaxation module implemented in the Webcoop QA system. As a cooperative QA system, Webcoop focuses on a “know how” component which directs users to other available options and presents neighbouring information in the absence of answers for the original query. The neighbouring information is extracted from a specially crafted knowledge base. However, as explained previously, the information in the underlying knowledge base must be associated with semantic information to support relaxation. Benamara and Saint-Dizier (2004) achieve this by implementing a hand crafted ontology for the knowledge base. This ontology describes the concepts and properties of the domain. When processing questions, if Webcoop is unable to derive an answer to the question, possible relaxation directions are presented to the user. Upon selection, Webcoop relaxes the original query and executes it on the knowledge base to present the new information to the user. In this way Webcoop manages the user misconception in query relaxation as the relaxation process is executed based on the user’s preferences. Benamara and Saint-Dizier (2004) define three relaxation methods that Webcoop employs:

- relaxation based on cardinality

- relaxation on the ontological type of the focus
- relaxation of constants

The relaxation based on cardinality concentrates on questions where the quantity of a resource does not comply with cardinality constraints mentioned in the question. For example, consider the question “*Which San Francisco hotel has rooms with five beds?*” where a particular user tries to find a five bed hotel room in San Francisco. However, assume that according to the information recorded in the knowledge base there are not any hotel rooms in San Francisco which have five bed rooms. In such scenarios, WEBCOOP relaxes the query, so that it searches hotel rooms with more than five beds. Such relaxation helps the user to find a hotel room that suits his/her needs in scenarios where a perfect match is unavailable.

If the ontological type of the question focus is available, then the query can be further relaxed by finding similar concepts through ontological relations. For example, instead of finding nearby hotels in the previous example, the QA system can also find chalet and pensions which are conceptually similar to the question focus. However, question parsing and an ontology resource which contains information on the concepts mentioned in the question are essential for this type of a relaxation.

Relaxation of the constants is carried out if the query failed due to a constant mentioned in the question. For example, consider the question “*What are the hotels located in Sandringham?*” which seeks hotels located only in Sandringham. However, if the query failed without results, then this query can be relaxed by relaxing the “*Sandringham*” to cities near Sandringham, which provides additional options to the user.

Benamara (2004) also reports the relaxation control strategies which are used in WEBCOOP. Since relaxation opens a wide variety of options for the QA system to present useful answers, there is a clear need for a control strategy to present only

useful answers to the user. Webcoop utilizes a supervising module which controls the relaxation by calling different relaxation modules individually and then allowing the system to sort them.

Clark (2010) reports the query relaxation approach which is used in the AURA QA system. Clark also presents three relaxation methods which are employed in AURA. These include:

- dropping a qualifier
- specializing the class in the antecedent
- generalizing a class in the consequent

If the QA system fails to answer a particular question due to an unrecognised noun modifier, then the relaxation process drops that qualifier and presents the retrieved information for the new query. For instance, if the question, “*who is the first president of America?*”, does not return any answer to the user, then the adjective “*first*” is dropped and the resulting text is formed as a new question. The resulting question will not present the required answer to the user, however, some related information can be presented. In addition, AURA also reports the failure of the original question while presenting this new information.

In the former relaxation operation, AURA drops the specialization by removing the adjectives. However, the inverse function of this process can also be carried out as a relaxation operation. The specializing class in the antecedent focuses on introducing new information to specialize a universally quantified class. For example, the question “*what religions are followed in India?*” can be relaxed by specializing “*India*” resulting a new question “*what religions are followed in South India?*”. As in the previous relaxation, the current relaxation can also present some interesting information that users are willing to see.

Table 2.7 Summary of the answer presentation approaches using alternative queries and query relaxation

Citation	Approach	Evaluation	Weaknesses
Benamara (2004)	Uses a handcrafted ontology and supports query relaxation based on cardinality, semantic relation between concepts included in the query and relaxation of literal values.	No specific evaluation is reported to find the effectiveness of the answer presentation model.	Requires an ontology of semantic relations between concepts that may be included in the query.
Clark (2010)	Relaxes queries by dropping qualifiers, specialising the class in the antecedent or generalising the class in the consequent.	Uses a 308 question set for the experiments. The quality and detailed accuracy of the relaxation is not measured except for some statistics on the number of questions relaxed by each of the three modules.	The relaxation models need a supervising module to determine when to relax a particular query. In some scenarios, a relaxation can present incorrect or less important information.

A summary of the answer presentation models discussed in this section is shown in Table 2.7. Alternative querying and query relaxation are key tools used to present answers with additional information which will be important for the knowledge seekers in the absence of the direct answer. The framework presented in this thesis concentrates on enriching the answer using information directly related to the entities mentioned in the answer and/or the question. On the other hand, query relaxation approaches do not focus on generating answers in natural language as introduced in this thesis with additional function to realize the generated answer.

2.4 User Tailored Answers

Answers can also be presented utilizing the user profile, where user preferences are explicitly acquired or derived using the interaction history with the system. In this section we discuss user tailored QA systems that focus on the presentation aspects of the answer, hence approaches discussed in this section represent a subset of the work carried out in user tailored QA.

Quarteroni (2010) reports the user model based QA approach which concentrates on the user's reading level and the topics of interest. The reading level component determines which age group can read the text. To accomplish this the model acquires the age group from the users and implements a multi-class classification on the text to assign it to the correct group. Quarteroni (2010) trained a multi-class classifier on a 180 document collection which was annotated by human participants by assigning one of the three labels; basic (7-11 age group), medium (11-16 age group), and advanced (all adults). Quarteroni (2010) uses the Smoothed Unigram Model for classification which is based on unigram language models. In addition, the unigrams are transformed into their base form by stemming through the Porter stemmer (Porter, 1980). The resulting classifier model is then used to classify new documents into one of the three reading

levels. Since the resulting model is based on the unigram features, it does not consider deep syntactic structures which affect the reading level of a text document. There is also the possibility that complex text can be written by combining simple words with advanced syntactic structures which are suitable only for the advanced reading level. However, Quarteroni (2010) does not concentrate on such scenarios which obviously present future directions in reading level analysis for answer presentation. Quarteroni (2010) also introduces the profile component which analyses the user's files and builds a keyword collection using the Kea (Witten et al., 1999) keyword extraction algorithm. This keyword collection represents the profile of a user which is then used in the QA process. In essence, if a user processes a question and retrieves a set of documents, instead of extracting the answer from the ranked document, a personalization of the answer is carried out by prioritizing the documents which contain the same keywords as ones that are stored in the user profile.

Walker et al. (2004) present user tailored responses in a dialogue system. Since QA systems are essentially dialogue systems, the user profiling and presentation techniques presented in this research can be directly employed for a QA system. They elicit the user models through a standard set of questions which are provided at the time of user registration. The answers provided for these questions are analysed to build the user models to be used for tailoring responses. A generated user model is basically used to decide what information should be presented to the user. In essence, the attributes recorded in the user model are used to filter the retrieved information from the system. Walker et al. (2004) concentrate on the restaurant domain to explain the user modelling scenario where users answer a set of questions which reveals their preference for the food quality, food type, cost, and other related attributes. There are two steps that occur when a particular user searches for a restaurant by providing a question. First the constraints that are mentioned in the question are consulted and then the user model

is consulted to filter restaurants based on the user preferences collected through the question set.

Allen et al. (2016) describe an approach towards answer presentation which is based on analysing the expertise level of the user. In this system, the answers are annotated by the system with the labels denoting novice, intermediate, or expert. The current demonstrations of this approach utilize questions which seek for answers containing a series of steps, each of which are annotated. When a particular user is seeking answers for the question, the model first checks the user profile of the user and then filters the information based on the user profile. For instance, if the user profile shows that he/she is an intermediate user, then all steps which are annotated as novice can be filtered out from the answer.

Several other QA systems (Liu and Agichtein, 2008; Thai et al., 2006; Zhang et al., 2006) also focus on generating personalized answers. However, these systems focus only on the answer filtering strategy which is covered in already discussed answer presentation models.

The user tailored QA systems present customized answers for the user. A summary of the user tailored answer presentation models discussed in this section is shown in Table 2.8. The user tailored QA systems, which incorporate additional information as in a model proposed by Walker et al. (2004), generate the answer based on the available user model. Although this approach may work for the users who search for information related to one of the main themes, it will not be suitable to generalize as users (i.e., knowledge seekers) may look for new information to expand their current understanding of the question. The framework, RealText, proposed in this thesis, focuses on this aspect and presents an answer which contains additional information related to the question/answer. This additional information transforms the bare factoid

Table 2.8 Summary of the answer presentation approaches using user tailored answers

Citation	Approach	Evaluation	Weaknesses
Quarteroni (2010)	Tailors responses using reading level and keywords. Users previous keyword patterns are used to prioritise documents, and reading level of the document is used to tailor the output.	A human evaluation using Likert Scale (scale of 1-5) is carried out to determine the usefulness, relatedness, time, and sensitivity. The usefulness (the factor that related to answer presentation directly) for the three queries which were included in the survey achieved satisfaction levels of 3.6, 2.3, and 3.3.	The user model is based on the Bag of Words model. This naive approach is not able to identify advanced semantic relationships.
Walker et al. (2004)	Collects data about the user on initial registration, and builds a user model which is used to filter responses.	Evaluation is carried out by human participants to determine tailoring of responses. The results indicates that users prefer tailored answers.	The system depends on the initial question set excessively. A dynamic approach of user model generation is required.
Allen et al. (2016)	Analyses the expertise level of the user and filters out unnecessary steps in the process that is the response to the user question (e.g., novice steps are filtered out for an intermediate user).	No specific evaluation is reported to find the effectiveness of the answer presentation model.	The model depends on the annotation of the steps. Sometimes a novice user may also prefer to see advanced level tasks when seeking additional knowledge.

answer into an *informative answer*, making this a good starting point for knowledge seekers.

2.5 Evaluating Answer Presentation

The evaluation of answer presentation can be discussed under two main themes; human evaluation and automatic metric based evaluation. Human evaluation concentrates on using human participants to rate answers or provide judgements on answers. The analysis of the results can be done either quantitatively or qualitatively based on the evaluation strategy. Automatic metrics evaluate answers based on the human reference answers and are compared with system generated answers. However, there are other automatic metrics that can provide quality judgements on answers.

The following sections discuss the two evaluation camps in detail. In some scenarios we also highlight some evaluation strategies that may be suitable for use in future research.

2.5.1 Human Evaluation

Human evaluation is the most widely used and also the de facto evaluation method for answer presentation. In general, a set of users rate the answer presentation mechanism based on a predetermined rating scale. If the QA system is based on a closed domain environment, then a set of experts in the domain are employed for evaluation.

2.5.1.1 Agreement on Common Evaluation Standard

The main requirement in human evaluation is that all users are required to agree on a common evaluation standard. Before starting the real evaluation the participants should be provided with training examples which show the basic rules of the evaluation.

It is also important to measure the inter-rater agreement once the evaluation is over. Cronbach's Alpha (Santos, 1999), Krippendorff's alpha (Krippendorff, 2004, 2007), or Fleiss Kappa (Fleiss, 1971) can be used to measure the inter-rater agreement, given that the result collection satisfies the requirements and constraints defined by the metric. These metrics provide an idea of the agreement between the different participants who have rated the same result set, and thus the reliability of the results.

In addition to the computation of agreement, it is important to analyse whether a significant change in the outcome of the result can occur if participants are removed from the group. In some scenarios participants may rate answer presentation schemas significantly different from others. To detect such outliers, it is important to analyse the inter-rater agreement by temporarily removing participants that may be outliers.

2.5.1.2 Rating Factors

The rating factors for human evaluation depend on the type of answers which are produced by the QA system. A textual answer presentation system can be evaluated on readability and accuracy factors. Accuracy can focus on the grammatical accuracy and the appropriateness of the presented answer. In some scenarios, a grammatically inaccurate answer can still be readable to humans. Therefore, the readability should be measured separately to the accuracy factor. Formally, accuracy covers the grammatical formation, spelling, and meaning conveyed in the generated text. The readability factor covers whether the text can be interpreted by a native speaker and adheres to the common use of the terms.

In certain scenarios, fluency and adequacy are also considered as the rating factors. These two factors also measure the same aspects measured by the accuracy and readability. Fluency measures grammatical formation, common use of terms, and the ability to be interpreted by a native speaker. Adequacy measures whether the meaning

is expressed in the generated text. When put together, adequacy and fluency measure the same aspects as readability and accuracy, although their individual focus is different. Therefore, an evaluation schema can focus on one of the two pairs to rate the generated text.

2.5.1.3 Drawbacks and Challenges in Human Evaluation

Although human evaluation can measure computer generated text more realistically than automated metrics, it has the following major drawbacks:

- *Resource cost:* Although most systems are evaluated with voluntary participants (e.g., Reiter and Belz (2009); Yu et al. (2007)), the cost of advertising and managing participants is usually expensive, and incentives may be required to encourage participation. This will generally include hiring a venue, refreshments, and other associated costs of managing the group of participants. In addition, if the answer presentation system is built for a closed domain environment, the evaluation requires domain experts as participants who would be even more difficult to approach and manage.
- *Difficulty in finding the same user group:* Answer presentation systems that adhere to the iterative development need to be tested on multiple occasions. However, it is not always possible to find the same group of participants who evaluated the system in the previous rounds. To solve this, the new group of participants need to be trained again which takes reasonable time in large scale application testing.

To overcome these drawbacks, many areas related to NLP are constantly searching for automatic metrics for evaluation which have high correlations with human judgements. In the next section we discuss the automatic metrics that can be utilized for textual answer presentation.

2.5.2 Automatic Metric based Evaluation

This section describes automatic metric based evaluation under three themes. The first two sections discuss automated accuracy and readability evaluation while the last section is devoted to a discussion on evaluation against human reference text using automatic metrics.

2.5.2.1 Automated Accuracy Evaluation

The automated evaluation of accuracy concentrates mainly on the grammatical and spelling accuracies of the presented answer. The main approach is to use a pre-trained or rule based grammar and spell checker to scan through the text and identify possible errors. For example, rule based grammar checking programs such as LanguageTool (Naber and Milkowski, 2016) and as well as commercial grammar checking tools such as Grammarly (Grammarly, 2016) are available to evaluate the text for grammatical correctness and spelling mistakes.

2.5.2.2 Automated Readability Evaluation

Automated readability evaluation focuses on whether the presented answer can be interpreted by a native speaker. This section introduces six different metrics that rank text for its readability.

The Automated Readability Index (ARI) (Smith and Senter, 1967) measures the readability of the text using the equation (2.1) shown below.

$$ARI = 4.71 \left(\frac{\text{characters}}{\text{words}} \right) + 0.5 \left(\frac{\text{words}}{\text{sentences}} \right) - 21.43 \quad (2.1)$$

This formula takes the words, characters, and the sentences to measure the readability without considering the syntactic structure of the sentence. However, the syntactic

structure plays a dominant role in readability and it is investigated in a number of different studies (Nilagupta, 1977; von Glasersfeld, 1970). Hence, ARI cannot be considered as a complete readability metric that can correctly measure an actual value of readability.

Flesch-Kincaid grade level (F-K) (Kincaid et al., 1975) is another readability metric which is similar to the ARI, except with a slight change in the attributes used. The grade level utilizes the words, sentences, and the syllables to measure the readability of the text as shown in the equation (2.2) below.

$$F - K \text{ grade} = 206.835 - 1.015 \left(\frac{\text{total words}}{\text{total sentences}} \right) - 84.6 \left(\frac{\text{total syllables}}{\text{total words}} \right) \quad (2.2)$$

Neither of the above two methods consider the appearance of complex words (Adams, 2014) to calculate the level of readability. Taking this factor into consideration, Gunning (1968) introduces the Gunning Fog index which takes the number of complex words into account when calculating the readability level of the text as shown in the equation (2.3) below.

$$\text{Gunning Fog index} = 0.4 \left[\left(\frac{\text{words}}{\text{sentences}} \right) + 100 \left(\frac{\text{complex words}}{\text{words}} \right) \right] \quad (2.3)$$

The main limitation of this approach is that it does not consider different senses of the same word. For example, the word “occupy” has 8 different synsets according to WordNet (Miller, 1995). These individual synsets explain different aspects of the same word and can contribute towards a complex meaning. This aspect is not considered in Gunning’s Fog index when calculating the complexity of the sentence based on word level features. In addition, as seen in earlier approaches, depending on the word level

features to estimate the readability of a sentence is mostly ineffective as readability of language is bound to the syntactic structure as well.

Several other metrics such as the SMOG index (McLaughlin, 1969), Fry readability formula (Fry, 1980), and Coleman–Liau index (Coleman and Liau, 1975) also use the sentences, words, and letters to calculate the readability level of a given text.

Although the aforementioned metrics are easy to calculate and provide scales to compare the readability level of a text, the key drawback is that the calculations are naive by nature and do not provide experimental evaluations. In essence, none of these metrics compare the readability of a text as determined by the metric with the readability determined by a human. However, instead of these naive approaches, a better alternative would be to compare the text with human provided reference text and assign a score based on the number of experiments. This sort of approach is widely used in the machine translation domain with a number of metrics. We discuss such approaches in the next section with an analysis on different metrics and the ability to employ them in the answer presentation domain.

2.5.2.3 Evaluating against Human Reference Text

Instead of evaluating presented answers based on metrics which are already set up with certain formula, there is also an opportunity to evaluate them by comparing them with a human generated answer. Although answer presentation has not employed this aspect so far, Machine Translation (MT) widely employs such automatic metrics to evaluate the translation quality of different systems. This discussion therefore focuses on using these metric in answer presentation.

This type of evaluation is initiated by producing a collection of human provided answers which are termed as human reference answers. Each system generated answer is then evaluated against the reference answer using an automatic metric. However,

an important factor to consider is how well the chosen metric values can represent the actual human judgements. This is generally analysed before the actual evaluation by computing the correlation between human scores provided for a sample test set and the metric generated scores for the same test set. In the discussion below we introduce 5 different automatic metrics which can be used to evaluate the answer presentation.

The Word Error Rate (WER) (Wang et al., 2003) focuses on evaluating the generated text by aligning it with a reference text and then analysing the substitutions, deletions, and inclusions as in the Levenshtein edit distance (Levenshtein, 1966). The equation (2.4) below illustrates the WER calculation strategy.

$$WER = \frac{S + D + I}{N} \quad (2.4)$$

Here, S , D , and I represent the substitutions, deletions and the insertions respectively while N represents the number of words in the reference. Based on the WER, accuracy ($WAcc$) can be calculated as $1 - WER$. Although WER provides an estimation of how the actual reference sentence deviates from the generated sentence, it does not provide any idea of actual errors that can exist in the generated answer.

The Longest Common Subsequence (LCS) measures the number of tokens that are aligned between the reference answer and the generated answer. According to Cormen et al. (1989) the formal definition of LCS is as follows. A sequence $X = \langle x_1, x_2, \dots, x_m \rangle$ is a subsequence of another sequence $Y = \langle y_1, y_2, \dots, y_n \rangle$, if there exists a strict increasing sequence $\langle i_1, i_2, \dots, i_k \rangle$ of indices of Y such that for all $j = 1, 2, \dots, k$, it satisfies $Y_{i_j} = X_j$. The common subsequence of X and Y with maximum length is then referred as the LCS. However, LCS does not require aligned tokens to appear in consecutive positions. Therefore, the semantic difference between the reference and the generated answer, is not properly taken into consideration. This is one

of the significant drawbacks in the LCS model when used to find the similarity between the human reference answer and the system generated answer.

Papineni et al. (2002) propose the BLEU metric which is based on the modified n-gram precision. A general unigram precision model simply counts up the occurrence count of candidate unigrams in reference and divides by the candidate word count. Instead, a modified unigram precision model first finds the maximum number of times a candidate unigram appears in any reference answer (*MaxReferenceCount*). Then the total count of each candidate token is clipped by the *MaxReferenceCount*, this ensures that the total count never exceeds the maximum reference count. The modified unigram precision can then be computed by dividing the sum of clipped counts by the total number of candidate tokens. This can be defined formally as in equation (2.5) below for a set of candidates.

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')} \quad (2.5)$$

Where $Count_{clip}$ can be defined as $\min(Count, MaxReferenceCount)$. According to the computation, the modified n-gram precision model penalizes the candidate answers which are longer than the reference answers. In addition, BLEU also comprises a multiplicative brevity penalty which enforces matching of length, word choice, and even word order between the candidate answer and reference answer.

ROUGE (Lin, 2004) is a package of metrics which are focused on evaluating a machine generated summary of text to a human provided reference summary. In this section we focus on ways of utilizing the ROUGE package in the answer presentation domain. The package contains four different metrics namely, ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N can be utilized to evaluate a machine generated answer against a human provided reference answer using the n-gram recall with the basis as in equation (2.6) below.

$$ROUGE - N = \frac{\sum_{S \in \{Reference\ Answers\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\ Answers\}} \sum_{gram_n \in S} Count(gram_n)} \quad (2.6)$$

Where n represents the length of the n-gram which is expressed as $gram_n$. The expression $Count_{match}(gram_n)$ represents the maximum number of n-grams co-occur in a reference answer and a generated answer. When applying ROUGE-N to answer presentation tasks, the most suitable n-gram count needs to be identified. Although this can be accomplished empirically based on a training data, there can be a large diversity among the answers which makes it difficult to find a generalizable n-gram count.

Meteor (Banerjee and Lavie, 2005; Denkowski and Lavie, 2011) is another MT evaluation metric that can be used to automate evaluation of generated answers using a reference set of human answers. Meteor is based on the unigram matching between the reference and the generated text. However, unlike BLEU which only focuses on the unigram precision, Meteor uses the unigram recall in the calculation of the final score. In addition to these two factors, Meteor also combines a fragmentation measure to decide the ordering of the matched words between the reference and the generated text. Meteor unigram matching is currently (as of version 1.3) composed of four different modules. This form of extensive mapping is important to consider when automatic metrics are employed in free-form text generation approaches as seen in answer presentation. Firstly, the unigram mapping module focuses on exact matches which reports which unigrams appear as is in the reference and generated answers. The stem matcher matches the words if the stems are identical using a Snowball stemmer (Porter, 2001) appropriate for the language of the sentences. Meteor also contains a synonym matcher which matches words if they share the membership of a synonym set which is retrieved from WordNet (Miller, 1995). The fourth matching module in Meteor uses paraphrases to match phrases. The paraphrase table is developed as a separate task

Hum \ Sys	the	mayor	of	Auckland	is	Phil	Goff
Phil						•	
Goff							•
is					•		
the	•						
mayor		•					
of			•				
Auckland				•			

(a) Phrase matching based on exact tokens

Hum \ Sys	the	husband	of	Melanija	Knava	is	Donald	Trump
Melanija				•				
Knava					•			
is						•		
married	•	•	•					
to	•	•	•					
Donald							•	
Trump								•

(b) Phrase matching based on a paraphrase table

Fig. 2.9 Two examples depicts the Meteor phrase matching technique

using the phrase table “pivoting” technique. For instance, Fig. 2.9a and Fig 2.9b depict two examples of Meteor phrase matching technique as utilized by Perera et al. (2017). It is evident from the figures that Meteor is capable enough to match similar tokens between the candidate and reference sentences, as well as to match multi-token phrases based on the paraphrase lexicon.

2.6 Chapter Summary

This chapter discussed answer presentation approaches that are related to the work reported in this thesis. We firstly presented a taxonomy to classify the literature based on the core technique used in the presentation. This resulted in three higher-level themes and some sub-themes which were discussed accordingly. In addition, we compared the work reported in this thesis with existing answer presentation models. The review also included a section devoted to discuss evaluation metrics which can be utilized to measure the accuracy and readability of generated answers, using both humans and automatic metrics.

The next chapter focuses on the methodology of the RealText, the framework proposed in this thesis to present answers with informative answers. The methodol-

ogy section will explain how the proposed framework advances the current answer presentation which is discussed in the current chapter.

Chapter 3

Methodology

This chapter describes the research methodology and the implementation of the RealText framework. This details the data source, the rationale and the design strategies including the implementation details.

The chapter is structured as follows. Section 3.1 explains the high-level architecture of the RealText framework. Sections 3.2 and 3.4 discuss the information sources utilized in the framework and question dataset respectively. Section 3.5 describes the answer sentence generation strategy followed by Sections 3.6, 3.7, and 3.8 which provide an in-depth analysis of the informative answer generated by the ensemble modules. We conclude with a summary of the chapter in Section 3.10.

3.1 RealText Architecture

A high-level architecture of the RealText framework is depicted in Fig. 3.1. The framework is based on a number of resources including Linked Data resources and an enhanced Linked Data based question dataset which is taken as the input.

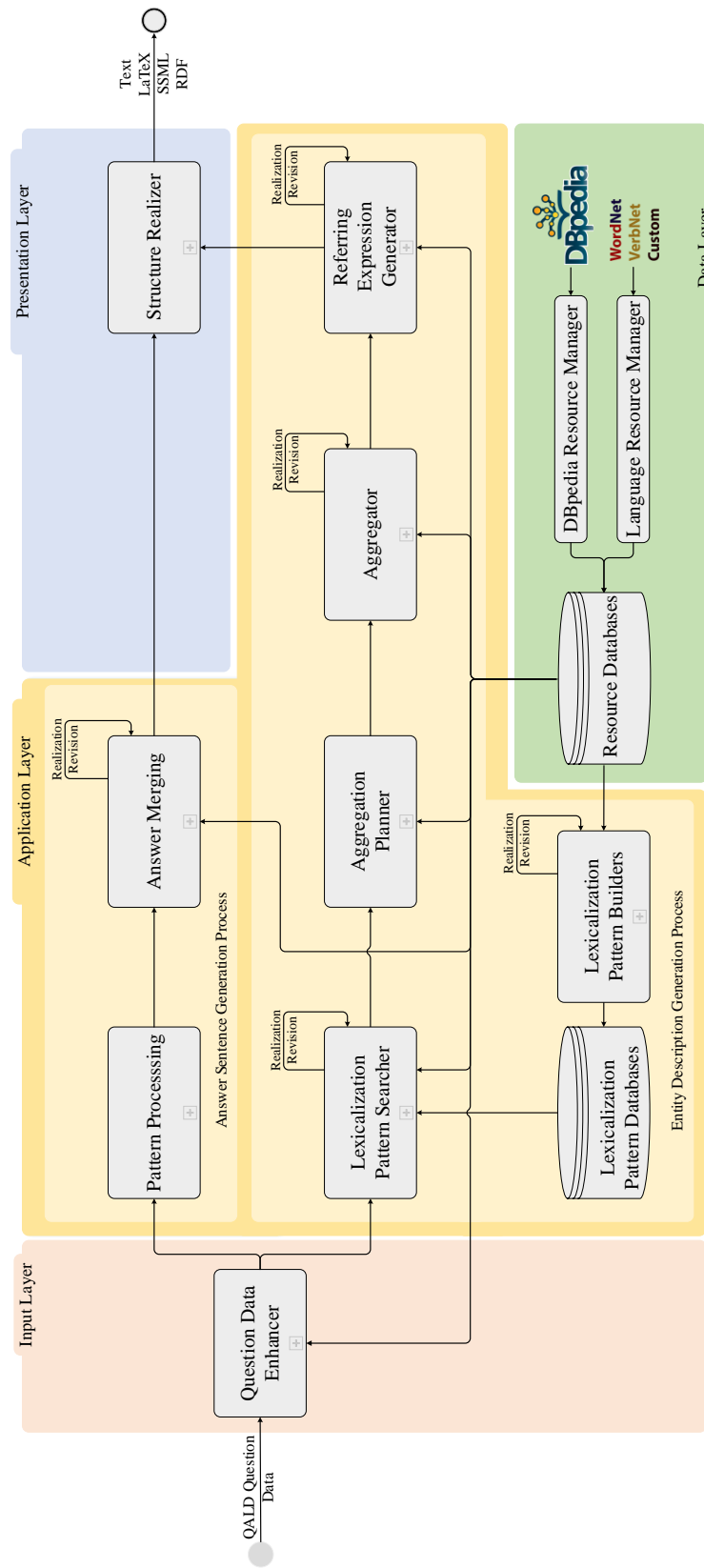


Fig. 3.1 High-level view of the RealText architecture

The framework is structured using four distinct layers, the *Data layer*, *Input layer*, *Application layer*, and *Presentation layer*.

The overall architecture is designed in a layered fashion due to a number of reasons in order to facilitate flexibility, maintainability, and scalability. This is achieved by firstly separating the answer sentence and the entity description generation into two layers, which is further divided into the input, application, data, and the presentation layer. This layered design allows the whole framework to independently evolve and scale as individual units. The challenge in such an approach is the communication protocol among various modules. In this research, we utilized the Attribute Value Matrices (AVM) mapped to Plain Old Java Objects (POJO) in the implementation for the communication between different modules. One of the key motivations behind the architecture design was to allow multiple applications to reuse the individual modules of the framework. For instance, certain external systems may be interested in using only the answer sentence generation module, which is easily possible with our stand alone design of the architecture. Any of the module can be used on its own by invoking a pre-defined AVM and the output will be generated in another pre-defined AVM which is consumable by the invoking module. We provide examples of AVMs later in this chapter when describing the various modules. In order to illustrate the overall functioning of RealText, we briefly describe below the individual modules and their cohesion with surrounding modules. Each of the modules are described in more detail later in this chapter.

The Data layer manages the information source utilized for the framework, in essence DBpedia and language resources (VerbNet, WordNet and custom built resources). As seen in the architecture diagram it is central to the overall framework and accessed in different modules with different granularities. Further details on how different components access data will be discussed when explaining the specific modules.

DBpedia online repository is not directly queried, instead the framework maintains an offline version of DBpedia which is wrapped by an application which can download Resource Description Framework (RDF) data on demand. This is due to three reasons; firstly the querying platform for DBpedia frequently goes out of service, secondly the online DBpedia SPARQL query endpoint is slow in query processing. The third reason targets on loose coupling DBpedia to the rest of the framework. In essence, this means utilizing a different Linked Data resource other than DBpedia, and it is also one of the future objectives of this research. DBpedia manager mediates between the DBpedia and the local databases by keeping a local copy of DBpedia RDF files and metadata databases which are required for the current question dataset. In addition, it can also download and store DBpedia RDF files on demand. This will be further discussed in Section 3.2. The framework also utilizes a series of language resources (both existing and custom) to infuse linguistic knowledge to the framework. Section 3.3 discusses these resources in detail. In addition to aforementioned resources, individual modules utilize additional databases. For the simplicity, we describe them when they first appear in the context.

Input layer comprises the input to the framework which is an enhanced question dataset based on QALD data. The data enhancing is carried out using the resources mentioned in the data layer. The structure of the input and the enhancing process will be discussed in Section 3.4. We also show an example of the enhanced input using AVM at the end of Section 3.4.

The Application layer manages the two main processes to build the informative answer, namely, answer sentence generation and entity description generation process. It is also worthwhile to note that certain modules carry out realization and revision in the module itself without using a certain sub-process. This deviates our architecture from the NLG consensus architecture proposed by Reiter and Dale (2000). The main benefit

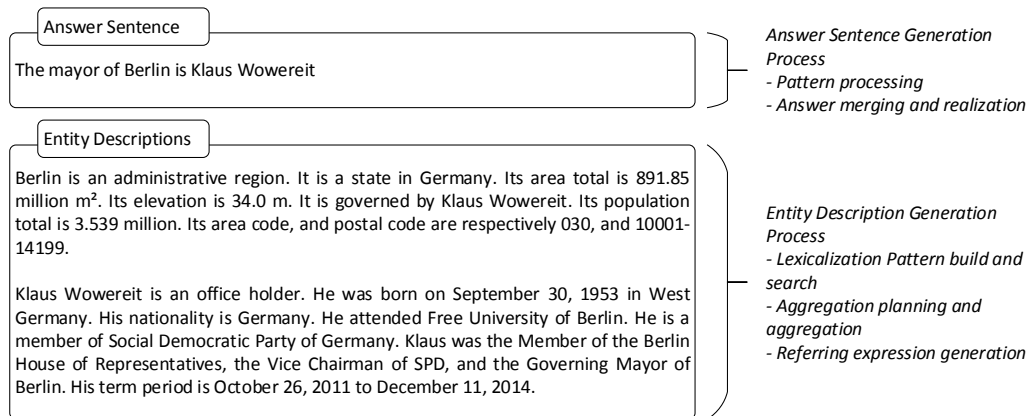


Fig. 3.2 Informative answer tagged with modules which contributed to its development. The answer is in response to the question “Who is the mayor of Berlin?” which has the answer “Klaus Wowereit”.

of this type of architecture is that the errors are identified using syntactic parsing and are fixed before they propagate to the latter modules. Figure 3.2 presents an example of an informative answer tagged with modules used to develop it. Section 3.5 discusses the answer sentence generation process as a whole including both pattern extraction and answer merging shown in Fig. 3.1. The entity description generation process contains multiple modules and will be explained as follows. In Section 3.6 we describe the lexicalization module. This covers both lexicalization pattern building and searching shown in Fig. 3.1. Section 3.7 elaborates the aggregation process which combines lexicalized triples into paragraphs. Both the aggregation planner and aggregator shown in Fig. 3.1 will be discussed in this section. In Section 3.8, we discuss the referring expression generation process.

The Presentation layer manages the presentation of the informative answer through structure realization. This layer has the functionality to transform the generated answer into six different presentation formats. Section 3.9 will explain the structure realization module of the framework that handles the presentation layer.

3.2 DBpedia as an Information Source

DBpedia is used as the information source of this framework due to a number of reasons. Firstly, DBpedia has become the central hub for the Linked Data hub (Auer et al., 2007; Kobilarov et al., 2009), and any solution that focuses on DBpedia contributes to the advancement of the whole Linked Data cloud. This is because the nature of Linked Data is based on the reuse of existing information by linking them appropriately from different resources. Secondly, the growth and interlinkage of DBpedia is high compared to other Linked Data resources, and importantly DBpedia information is structured under well designed ontology class hierarchy. Another reason is that open domain QALD datasets such as one mentioned in this research (will be explained in detail in Section 3.4) tend to use DBpedia as the main information source due to its open domain nature and high coverage.

In the following sections we describe the role of DBpedia as an information source in the framework. Section 3.2.1 provides an overview of DBpedia structure. Section 3.2.2 discusses DBpedia content coverage compared to other related Linked Data resources, its growth over recent releases, and incoming and outgoing links which make it into a interlinking hub for Linked Data thus making a good information source.

3.2.1 Structure of DBpedia

DBpedia is currently the leading open domain Linked Data resource which extracts information from Wikipedia and transforms them into Linked Data. It is often termed as the interlinking hub for Linked Data (Bizer et al., 2009; Kobilarov et al., 2009). The DBpedia information extraction process extracts information mentioned in Wikipedia infoboxes and converts them into triple form. These triples are stored in separate RDF files with the Wikipedia article name as the subject. For instance, triples related to Steve Jobs's Wikipedia article (https://en.wikipedia.org/wiki/Steve_Jobs) can be found in the

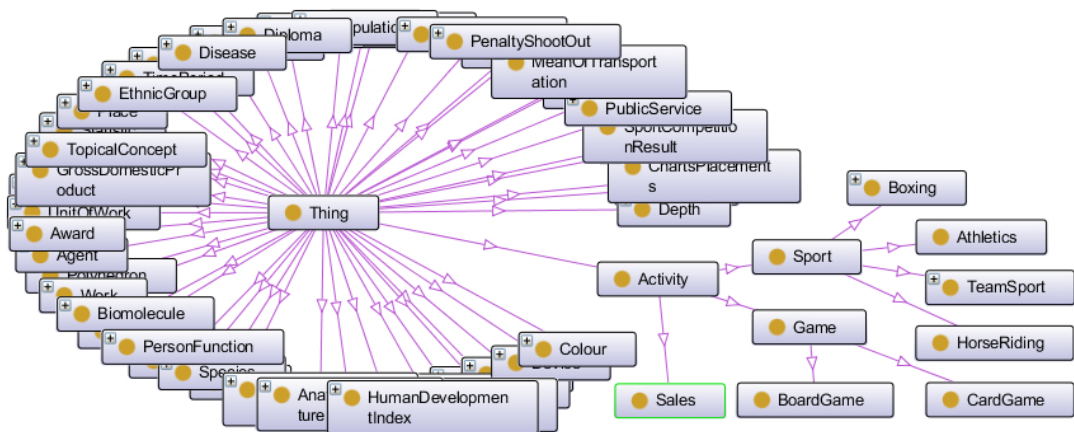


Fig. 3.3 DBpedia ontology class visualization

corresponding DBpedia entity (http://dbpedia.org/page/Steve_Jobs). These DBpedia entities are organized under an ontology which currently contains 739 ontology classes.

DBpedia offers data in 13 different formats including RDF/XML, N-Triples and Turtle. The current research utilizes the RDF/XML format which is widely used in the Semantic Web applications. We have manually generated the Web Ontology Language (OWL) specification of DBpedia class hierarchy as mentioned in the DBpedia ontology class specification¹. Figure 3.3 depicts a visualization of the current ontology.

The current DBpedia versions contain two types of triples, DBpedia-OWL schema triples and the DBpedia properties. The latter is the traditional way of representing DBpedia triples which uses a set of names extracted from the Wikipedia Infoboxes. The main drawback in this approach is that same predicate is shown in multiple different ways. However, the DBpedia mapping project (Lehmann et al., 2014) focused on solving this issue by mapping already extracted properties to an ontology schema to standardize the information resulting in DBpedia-OWL schema triples. We utilize the DBpedia-OWL schema triples in this research.

¹<http://mappings.dbpedia.org/server/ontology/classes/>

3.2.2 DBpedia Suitability for QA Systems

DBpedia has become a key resource for QA systems with the recent trend towards the QALD and there are two main reasons which cause DBpedia to become the main Linked Data resource for QALD. Firstly, DBpedia has a very high growth rate which has not compromised its quality. This supports any QA system to benefit from large structured information source from which answers can be extracted with ease and with a high reliability. Secondly, in addition to interlinking its own triples, DBpedia is densely interlinked with other heterogeneous Linked Data resources making it the interlinking hub for the ever-growing Linked Data cloud (Kobilarov et al., 2009). Therefore, any system that focuses on DBpedia ultimately contributes to the whole Linked Data cloud. Sections 3.2.2.1 and 3.2.2.2 discuss the growth and interlinking of DBpedia in detail and provide comparisons with other Linked Data resources where necessary.

3.2.2.1 Growth of DBpedia

DBpedia has shown the highest growth rate among other open domain Linked Data resources. Table 3.1 shows a comparison between DBpedia and four leading Linked Data resources, Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007), Wikidata (Erxleben et al., 2014; Vrandečić and Krötzsch, 2014), and OpenCyc (Matuszek et al., 2006). Compared to other Linked Data resources, DBpedia contains more triples categorized under well organized class hierarchy and entities, which make it a strong choice as an information source for open domain QA. Although, other resources have a higher number of ontology classes, they do not have as many triples raising the issue of data sparsity. In addition, Yago, Wikidata, and OpenCyc are still in their initial phases of archiving Linked Data compared to DBpedia.

Table 3.2 reports the entities, triples and ontology classes in the last seven releases of DBpedia. It is clear according to the statistics that DBpedia has grown rapidly during

Table 3.1 Comparison of DBpedia statistics with Freebase, Yago, Wikidata, and OpenCyc. Statistics are taken from release notes and analysis reported by Färber et al. (2016). Entities per ontology class is shown in EPC column.

Linked Data Resource	Entities (millions)	Triples (billions)	Ontology classes	Entities per Class	Query language
DBpedia	6.2	4.3	739	8389.71	SPARQL
Freebase	44	2.4	40616	1083.31	MQL
YAGO	10	0.12	350000	28.57	SPARQL
Wikidata	18	0.740	302280	59.54	SPARQL
OpenCyc	0.04	0.002	116822	0.35	SPARQL

Table 3.2 DBpedia growth rate in last seven releases. Only the number of entities, triples and ontology classes are considered in the English edition of DBpedia.

Release version	Entities (millions)	Triples (billions)	Ontology classes
2015(b)	6.2	4.3	739
2015(a)	5.9	3.13	735
2014	4.58	3	685
3.9	4.26	2.46	529
3.8	3.77	1.89	359
3.7	3.64	1	320
3.6	3.5	0.672	272

the last few releases, making it into one of the fastest growing and massive Linked Data resource which is also open domain and free to use.

3.2.2.2 DBpedia Interlinking across Heterogeneous Linked Data Sources

In addition to linking triples within one data source, Linked Data also supports linking multiple heterogeneous data sources with the same underlying linking mechanism as described in Chapter 1. This enables all Linked Data resources to reside in a connected cloud with shared information. In this connected cloud, DBpedia plays a significant role as the central interlinking hub as it is the most densely interlinked Linked Data resource (Bizer, 2009; Kobilarov et al., 2009).

Table 3.3 Statistics on DBpedia interlinking

	Incoming links	Outgoing links
Total links	39 million	4.9 million
Number of datasets	181	14
Top 5 resources	i. Linked Open Colours ii. DBpedia Lite iii. Flickr Wrapper iv. Freebase v. YAGO	i. Freebase ii. Flickr Wrapper iii. WordNet iv. GeoNames v. UMBEL

3.2.3 DBpedia Databases

We have built several databases which integrate DBpedia related information to facilitate the framework. This step has two objectives. Firstly, it reduces the computational time as RDF parsing is a time and resource expensive task hence fetching the information and parsing the information in real-time compromises the speed of the system. Secondly, the intention is to build an intermediate layer between DBpedia and the framework. This is specifically to mitigate the effect on our system from any schema change that might occur on the DBpedia side.

3.2.3.1 Gender Database

The gender database contains the grammatical gender of people mentioned in the DBpedia. This database is based on the DBpedia grammatical gender NLP dataset including some missing records which were populated manually. Table 3.4 shows a sample set of selected records from the database. Additionally, the framework is equipped with a SPARQL querying mechanism to query DBpedia to find the grammatical gender of the entity being searched if it is not present in the database. This service uses the SPARQL mentioned in the Listing 3.1 and it is only triggered if the gender property is specified in the DBpedia as a property for the requested entity.

Table 3.4 Sample set of records from the gender database

DBpedia entity	Gender
Eddy_King	male
John_Picacio	male
Sansanee_Wattananukul	female
Michaela_Breeze	female
Christopher_Loria	male
Dwayne_Leik	male

```

1 PREFIX dbo: <http://dbpedia.org/ontology/>
2 PREFIX res: <http://dbpedia.org/resource/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 SELECT ?g1 WHERE
5 {
6   res:ENTITY_NAME dbo:gender ?g.
7   ?g rdfs:label ?g1.
8   FILTER(lang(?g1) = "en").
9 }

```

Listing 3.1 SPARQL query to retrieve gender of an entity

3.2.3.2 Unit of Measurement Database

The current data representation in DBpedia provides only the XML schema definition with the predicate which represents numerical (e.g., double, integer) or temporal (e.g., date/time) properties. The predicates which require measurement units in the real world are not associated with the unit of measurement. This becomes a real challenge when transforming these predicates into natural language. For example, to transform the triple $\langle \text{Michael Jordan}, \text{height}, 1.98 \rangle_T$ to natural language, we need the unit of measurement for height. To address this problem, a unit of measurement database is created which provides details on predicates which require the unit of measurement. Table 3.5 depicts sample set of selected records from this database.

Table 3.5 Sample set of records from the measurement unit database. The BASE_URI is equivalent to <http://dbpedia.org/ontology>.

Predicate	Ontology URI	Unit Longname	Unit Shortname
height	BASE_URI/height	meter	m
areaTotal	BASE_URI/areaTotal	square meter	m ²
discharge	BASE_URI/discharge	cubic meter	m ³
netIncome	BASE_URI/netIncome	US Dollars	USD
weight	BASE_URI/weight	gram	g

Table 3.6 Sample set of records from the ontology class – predicate database. The records were extracted from the ontology class hierarchy “*Agent, Person, Artist*”.

DBpedia Hierarchy	Predicate	Priority
Agent, Person, Artist	alias	1
Agent, Person, Artist	nationality	2
Agent, Person, Artist	birthDate	3
Agent, Person, Artist	birthPlace	4
Agent, Person, Artist	field	5
Agent, Person, Artist	movement	6
Agent, Person, Artist	deathDate	7
Agent, Person, Artist	deathPlace	8

3.2.3.3 Ontology Class Predicate Database

The ontology class predicate database contains records on predicates associated with DBpedia ontology class. Although it is possible to derive these from DBpedia OWL specification, this task is quite resource heavy due to parsing. The indexed embedded database provides flexibility for adapting the framework to schema changes under minimal modifications. Furthermore, the predicates are not associated with the priorities in the DBpedia OWL specification. However, the priority of a predicate is an important factor when generating natural language because the triples have to appear in appropriate positions in the generated text. To address this, we have included the priority value for each predicate based on the Wikipedia data. Table 3.6 depicts a sample set of selected records from this database.

Table 3.7 Sample set of records from the ontology class – entity database. The records were extracted from the ontology class “*Place, PopulatedPlace, Region, AdministrativeRegion*”.

DBpedia Hierarchy	Resource
Place,PopulatedPlace,Region,AdministrativeRegion	Alaska
Place,PopulatedPlace,Region,AdministrativeRegion	Berlin
Place,PopulatedPlace,Region,AdministrativeRegion	Hawaii
Place,PopulatedPlace,Region,AdministrativeRegion	Minnesota
Place,PopulatedPlace,Region,AdministrativeRegion	Nevada
Place,PopulatedPlace,Region,AdministrativeRegion	Texas
Place,PopulatedPlace,Region,AdministrativeRegion	Wisconsin

3.2.3.4 Ontology Class Entity Database

This database contains the associations between ontology classes and the entities. Although it is possible to extract the ontology classes from the entity RDF file, these classes are not hierarchically organized based on the DBpedia ontology class hierarchy. This creates a need for querying the DBpedia ontology OWL specification and to organize them in an easily accessible hierarchy. This database was developed to eliminate the burden of intermediate RDF parsing and querying leading to a loss in efficiency. Additionally, it provides a level of abstraction between DBpedia and the proposed framework. Table 3.7 reports a sample set of selected records from this database.

3.2.3.5 Predicate Date Database

Although DBpedia provides the XML schema definition with predicates to identify them as dates, this lacks the information to identify the type of date (i.e., whether it is a year, day and month, or any other combination).

The predicate date database contains the mappings from predicates which require temporal information to the type of the temporal information. This database is built using a semi-supervised approach. We first crawled DBpedia entities and identified the

Table 3.8 Sample set of records from the predicate – date database

Predicate	Type	Format
birthDate	Single	YMD
deathDate	Single	YMD
formationDate	Single	YMD
birthYear	Single	Y
deathYear	Single	Y
ethnicGroupsInYear	Single	Y

Table 3.9 Sample set of records from the predicate number database

Predicates that require numericals			
populationTotal	numberOfStudents	wins	championships
poles	races	podiums	numberOfEpisodes
numberOfEmployees	populationUrban	vehiclesPerDay	populationMetro

predicates which associate XML schema definition of date types by parsing the RDF files. Next, we manually analysed the entities and recorded the predicates which need partial date/time values as objects. A sample set of selected records from this database is shown in Table 3.8.

3.2.3.6 Predicate Number Database

The predicate number database stores the predicates which require a numerical value as the object value. This database does not include records of the unit of measurement database mentioned in Section 3.2.3.2, and instead only the objects with pure numerical values are recorded. The measured numericals and normal numerical values are treated separately as they are subjected to different verbalization approaches. A sample set of selected records from this database is shown in Table 3.9.

3.3 Language Resources

The framework utilizes two linguistic resources to infuse the language knowledge into different stages. The applicability of these resources will be discussed in the modules which utilize these resources. This section provides an initial introduction to the resources.

3.3.1 Verb Information Database

The verb information database is a modification of VerbNet (Kipper et al., 2008) lexicon. Table 3.10 depicts a sample set of selected records from this database. The database currently contains 3773 records.

VerbNet provides verb base forms and does not contain the inflection of the verbs based on the grammar. We used both the SimpleNLG (Gatt and Reiter, 2009) and DictService (O’Neill, 2011) to get the required verb inflections. Each verb in base form is associated with the past tense, past participle form, progressive form, and the third person singular form. Furthermore, VerbNet classifies the frames related to a verb, however, we have aggregated them under verb base form to check the existence of a frame given the verb.

3.3.2 Masculine-Feminine Token Database

This database contains the English tokens classified into masculine and feminine categories. Table 3.11 depicts a sample set of selected records from this database.

Table 3.10 Verb information database

Base form	Past tense	Past participle	Progressive form	Third person singular	Frame types (Comma separated)
abridge	abridged	abridged	abridging	abridges	NP.patient V, NP V NP.patient, NP.instrument V NP, NP V ADV- Middle, NP V NP PP.instrument
accept	accepted	accepted	accepting	accepts	NP V NP, NP V what S, NP V how S, NP V that S, NP V S_ING, NP V NP PP.source, NP V NP
accumulate	accumulated	accumulated	accumulating	accumulates	NP V NP together, NP V NP.theme, NP V, NP V NP PP.location, NP V NP
activate	activated	activated	activating	activates	NP.patient V, NP V NP.patient, NP.instrument V NP, NP V ADV- Middle, NP V NP PP.instrument
advertise	advertised	advertised	advertising	advertises	NP V PP.location, NP V PP.theme NP.location, NP V PP.location PP.theme, NP V NP PP.theme

Table 3.11 Sample set of records from the masculine – feminine token database

Masculine Token	Feminine Token
actor	actress
author	authoress
bachelor	spinster
boy	girl
husband	wife

3.4 Question Dataset

Since the proposed framework works on the answer presentation stage, the input contains both the question and factoid answer. In Semantic Web based QA, the question is transformed into a SPARQL query which is the standard querying mechanism for Semantic Web. Our initial question set, factoid questions extracted from QALD, contained the following three factors; the question, answer, and the SPARQL query. The statistics related to the question set will be further explained in Chapter 4. This section will only focus on structure of the question set which is needed to follow up with rest of the sections.

We first executed an enhancing process as a preprocessing task on the question dataset to support latter processes. The enhancing tasks focused on identifying the entities mentioned in the question, extracting triples of the identified entities, and answer type classification. Section 3.4.1, Section 3.4.3, and Section 3.4.2 explain the three enhancing tasks. Finally, Section 3.4.4 explains the structure of the enhanced question dataset which is provided as the main input to the system.

3.4.1 Entity Extraction

The framework needs all entities mentioned in the question to generate entity descriptions by extracting triples from the DBpedia resources related to mentioned entities. Given the SPARQL query, we parse it and extract triple patterns from the graph. A

sample SPARQL query from the dataset and algebraic expression of this query which shows the triple patterns mentioned in the query are shown respectively in Listing 3.2 and Listing 3.3.

```

1 PREFIX dbo: <http://dbpedia.org/ontology/>
2 PREFIX res: <http://dbpedia.org/resource/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 SELECT DISTINCT ?uri ?string WHERE
5 {
6   res:John_F._Kennedy dbo:successor ?uri .
7   OPTIONAL { ?uri rdfs:label ?string . FILTER (lang(?string) = 'en')
8             }
9 }

```

Listing 3.2 SPARQL query for the question “Who was the successor of John F. Kennedy?”

```

1 ( distinct
2   ( project (?uri ?string)
3     ( leftjoin
4       ( bgp ( triple <http://dbpedia.org/resource/John_F._Kennedy> <
5         http://dbpedia.org/ontology/successor> ?uri))
6       ( bgp ( triple ?uri <http://www.w3.org/2000/01/rdf-schema#label>
7         ?string))      (= (lang ?string) "en")))))

```

Listing 3.3 SPARQL algebraic expression of the query shown in Listing 3.2. Lines 4 and 5 shows the triple patterns mentioned in the query.

The framework iterates through the complete list of triple patterns and extracts ones which points to DBpedia resources. These entities are then used to extract the triples from the DBpedia using the process mentioned in Section 3.4.2.

3.4.2 Triple Extraction and Metadata Embedding

The entities identified in Section 3.4.1 are used to initiate the triple extraction process. Since we maintain a local copy of the DBpedia as mentioned in the Section 3.1, the triple extraction process focused on parsing RDF files to retrieve the triples. The RDF parsing is carried out through Jena toolkit (McBride, 2002). We also filtered out the triples which are not appropriate to the framework. These included triples that contain identifiers (e.g., VIAF ID) and that contain outgoing links and image URLs (e.g., Wikipedia page URL, image URL).

The triples are then embedded with metadata required to support the latter modules. The metadata include the properties that are determined using the DBpedia databases mentioned in Section 3.2.3 and two new information units; multiplicity of triple object and the verbalization of the triple. The multiplicity of the triple object focused on whether multiple triples exist in the collection with the same subject and predicate, but with different objects. This feature is used in the lexicalization process and more information on its usage can be found in Section 3.6.4. The verbalization process transforms the triples into natural language equivalent forms using a predefined rule set. This includes removing language identifiers (e.g., @en), verbalizing entity names (e.g., `Marlon_Fernandez_(Footballer_born_2001)` \Rightarrow Marlon Fernandez), and verbalizing object values to support the lexicalization module. The object value verbalization focuses on transforming date values into 7 different formats, representing measured variables in different units (e.g., 1.3 m, 130cm), and representing normal numbers in different scales and in verbal form. The application of these verbalizations will be further discussed in Section 3.6.4.4.

Table 3.12 Answer type classification

Type	Description	Classification Method
Boolean	True or false values	SPARQL based
Numeric	Any numerical value	Schema based
Date	Date value or date ranges	Schema based
String	Any text representation	None

3.4.3 Answer Type Classification and Verbalization

The type of answer must be identified to support the framework modules and this, especially is one of the main requirements for the answer sentence generation module (see Section 3.5). We classified answers into four different categories as shown in the first column of Table 3.12. The third column of the same table shows the method used to classify the questions.

We first classified the questions which seek for boolean answers using SPARQL query. From the four types of SPARQL queries (SELECT, CONSTRUCT, ASK, DESCRIBE), only SELECT and ASK queries can be presented in factoid question answering. Others are used to extract information and transform to RDF (CONSTRUCT) or are used to extract an RDF graph (DESCRIBE). The ASK queries can only be used to extract a boolean value from the RDF graph, thus making the answer type a boolean value. However the SELECT queries are used to extract any other value and cannot differentiate between the rest of the three types (numeric, date, string) of answers.

The numeric and date types are identified using the XML schema definition associated with the answer. The framework contains a DBpedia querying module over the web and all queries are executed on the DBpedia and answers are extracted. We then identify the data type from the XML schema definition and associate them with the necessary answer type as shown in Listing 3.4 (e.g., `?height = "1.81"^^xsd:double` \implies numeric). The rest of the answers are assigned with string data type.

```
1 PREFIX res: <http://dbpedia.org/resource/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 SELECT DISTINCT ?height WHERE
4 {
5   res:Claudia_Schiffer dbo:height ?height .
6 }
7 Result: ( ?height = "1.81"^^xsd:double ) ⇒ Numeric
```

Listing 3.4 Identifying answer type using XML schema definitions associated with SPARQL query result

The answers extracted from the Linked Data are verbalized as they include URL friendly characters (Steve_Jobs → Steve Jobs) and date values are verbalized into multiple formats (e.g., May 25, 2017) to support the rest of the modules.

3.4.4 Structure of the Enhanced Question Dataset

We embedded the new information identified through the aforementioned enhancing tasks to the question forming an enhanced question which is provided to the rest of the modules. The enhanced question structure is depicted in Fig. 3.5.

3.5 Answer Sentence Generation

The informative answer generated from the RealText framework is comprised of two main components, the answer sentence and a collection of entity descriptions (as described in Section 3.1). This section focuses on the answer sentence generation process in which the syntactic structure of the source question is utilized to generate a natural language sentence which embeds the factoid answer, thus helping the framework to present the factoid answer in a more human-friendly manner. Section 3.5.1 reports the

QuestionID	1																																																	
QuestionText	Who was the successor of John F. Kennedy?																																																	
Answer	Lyndon B. Johnson																																																	
AnswerType	String																																																	
DBpediaLinks	1	<table border="1"> <tr><td>Link:</td><td>http://dbpedia.org/resource/John_F._Kennedy</td></tr> <tr><td>Source:</td><td>Question</td></tr> <tr><td>Selected:</td><td>True</td></tr> </table>	Link:	http://dbpedia.org/resource/John_F._Kennedy	Source:	Question	Selected:	True																																										
	Link:	http://dbpedia.org/resource/John_F._Kennedy																																																
Source:	Question																																																	
Selected:	True																																																	
2	<table border="1"> <tr><td>Link:</td><td>http://dbpedia.org/resource/Lyndon_B._Johnson</td></tr> <tr><td>Source:</td><td>Answer</td></tr> <tr><td>Selected:</td><td>True</td></tr> </table>	Link:	http://dbpedia.org/resource/Lyndon_B._Johnson	Source:	Answer	Selected:	True																																											
Link:	http://dbpedia.org/resource/Lyndon_B._Johnson																																																	
Source:	Answer																																																	
Selected:	True																																																	
SPARQL	SELECT DISTINCT ?uri ?string WHERE ...																																																	
Triples	1	<table border="1"> <tr><td>Subject_{Raw}</td><td>John_F._Kennedy</td></tr> <tr><td>Predicate_{Raw}</td><td>birthDate</td></tr> <tr><td>Object_{Raw}</td><td>1917-05-29</td></tr> <tr><td>Subject_{Verbalized}</td><td>John F. Kennedy</td></tr> <tr><td>Predicate_{Verbalized}</td><td>birth date</td></tr> <tr><td>Object_{Verbalized}</td><td> <table border="1"> <tr><td>1</td><td>May 29, 1917</td></tr> <tr><td>2</td><td>29 May 1917</td></tr> <tr><td>3</td><td>.....</td></tr> </table> </td></tr> <tr><td>OntologyClasses</td><td> <table border="1"> <tr><td>1</td><td>Agent</td></tr> <tr><td>2</td><td>Person</td></tr> <tr><td>3</td><td>Office Holder</td></tr> </table> </td></tr> <tr><td>Predicate(RequireDate)</td><td>True</td></tr> <tr><td>Predicate(DateInfo)</td><td> <table border="1"> <tr><td>Type</td><td>Single</td></tr> <tr><td>Format</td><td>YMD</td></tr> </table> </td></tr> <tr><td>Predicate(RequireNormalNumber)</td><td>False</td></tr> <tr><td>Predicate(RequireMeasuredNumber)</td><td>False</td></tr> <tr><td>Predicate(MeasurementUnitInfo)</td><td> <table border="1"> <tr><td>Short name</td><td>Null</td></tr> <tr><td>Long name</td><td>Null</td></tr> </table> </td></tr> <tr><td>NaturalGender</td><td>Male</td></tr> <tr><td>Multiplicity</td><td>False</td></tr> </table>	Subject _{Raw}	John_F._Kennedy	Predicate _{Raw}	birthDate	Object _{Raw}	1917-05-29	Subject _{Verbalized}	John F. Kennedy	Predicate _{Verbalized}	birth date	Object _{Verbalized}	<table border="1"> <tr><td>1</td><td>May 29, 1917</td></tr> <tr><td>2</td><td>29 May 1917</td></tr> <tr><td>3</td><td>.....</td></tr> </table>	1	May 29, 1917	2	29 May 1917	3	OntologyClasses	<table border="1"> <tr><td>1</td><td>Agent</td></tr> <tr><td>2</td><td>Person</td></tr> <tr><td>3</td><td>Office Holder</td></tr> </table>	1	Agent	2	Person	3	Office Holder	Predicate(RequireDate)	True	Predicate(DateInfo)	<table border="1"> <tr><td>Type</td><td>Single</td></tr> <tr><td>Format</td><td>YMD</td></tr> </table>	Type	Single	Format	YMD	Predicate(RequireNormalNumber)	False	Predicate(RequireMeasuredNumber)	False	Predicate(MeasurementUnitInfo)	<table border="1"> <tr><td>Short name</td><td>Null</td></tr> <tr><td>Long name</td><td>Null</td></tr> </table>	Short name	Null	Long name	Null	NaturalGender	Male	Multiplicity	False
	Subject _{Raw}	John_F._Kennedy																																																
Predicate _{Raw}	birthDate																																																	
Object _{Raw}	1917-05-29																																																	
Subject _{Verbalized}	John F. Kennedy																																																	
Predicate _{Verbalized}	birth date																																																	
Object _{Verbalized}	<table border="1"> <tr><td>1</td><td>May 29, 1917</td></tr> <tr><td>2</td><td>29 May 1917</td></tr> <tr><td>3</td><td>.....</td></tr> </table>	1	May 29, 1917	2	29 May 1917	3																																											
1	May 29, 1917																																																	
2	29 May 1917																																																	
3																																																	
OntologyClasses	<table border="1"> <tr><td>1</td><td>Agent</td></tr> <tr><td>2</td><td>Person</td></tr> <tr><td>3</td><td>Office Holder</td></tr> </table>	1	Agent	2	Person	3	Office Holder																																											
1	Agent																																																	
2	Person																																																	
3	Office Holder																																																	
Predicate(RequireDate)	True																																																	
Predicate(DateInfo)	<table border="1"> <tr><td>Type</td><td>Single</td></tr> <tr><td>Format</td><td>YMD</td></tr> </table>	Type	Single	Format	YMD																																													
Type	Single																																																	
Format	YMD																																																	
Predicate(RequireNormalNumber)	False																																																	
Predicate(RequireMeasuredNumber)	False																																																	
Predicate(MeasurementUnitInfo)	<table border="1"> <tr><td>Short name</td><td>Null</td></tr> <tr><td>Long name</td><td>Null</td></tr> </table>	Short name	Null	Long name	Null																																													
Short name	Null																																																	
Long name	Null																																																	
NaturalGender	Male																																																	
Multiplicity	False																																																	
2	[.....]																																																	

Fig. 3.5 A sample attribute value matrix for an enhanced question with newly derived information

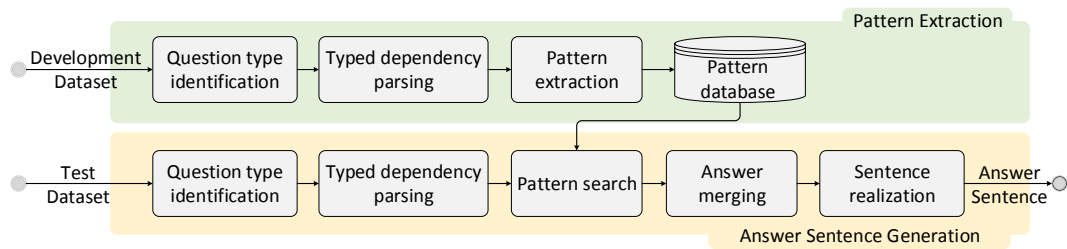


Fig. 3.6 Schematic representation of the answer sentence generation process

overview of the answer sentence generation process, and Section 3.5.2 to Section 3.5.6 are devoted to detailed discussion of different modules utilized in the process.

3.5.1 Answer Sentence and Process Overview

Factoid QA systems are generally designed to output the answer as a single piece of information. Although this is sufficient, it is very machine-like, hence there is a need to present answers which are more human friendly. This research investigated ways of presenting a factoid answer in a human-like sentence, built using the linguistic structure of the source question. The resulting answer sentence is a full sentence in natural language which contains the direct answer to the question.

We employed the typed dependency parsing to determine the linguistic structure of the source question. The core idea in this approach is to identify linguistic patterns using the typed dependency pattern of the source question and implement answer merging and realization mechanisms for the identified patterns. Any new question and answer pairs can then be realized into full answer sentences by using these known patterns and by applying associated merging and realization mechanisms.

Figure 3.6 depicts the schematic representation of the answer sentence generation process. In the following sections we first describe the question type identification process and then proceed to a detailed discussion on individual modules of the process.

3.5.2 Question Type Identification

It is essential to understand the type of the question to successfully generate an answer sentence with an embedded answer. Questions can be classified into two main types, wh-interrogatives and polar interrogatives (Ginzburg and Sag, 2000; Heycock, 2014). A wh-interrogative aims at getting an answer which represents another entity or a property of a resource mentioned in the question. On the other hand, polar interrogatives require true/false answers based on the statement presented in the question. These two question types need different answer sentence generation schemes. In wh-interrogatives, the answer must be embedded into the question linguistic structure by modifying the linguistic structure of the question, while polar interrogatives do not require such an embedding although they need modification of the linguistic structure of the question.

Since the answer sentence generation process depends on the question type, it is vital to classify the questions based on the interrogative type before extracting typed dependency patterns. As the current research concentrates on answer presentation which is the last step of the QA process, we exploited both the question and query to classify the questions into the correct interrogative type. We first classified all questions which require boolean value answers (as identified in Section 3.4.3) as polar interrogatives. The rest can be classified as wh-interrogative. However, to further validate this approach, the question text is Part of Speech (POS) tagged and analysed whether they contain the required POS tags. Table 3.13 depicts the required POS tags for wh- and polar interrogatives.

It is important to notice that in this research we do not consider imperative constructs which are not based on interrogative words, and such a construct is called a *statement* and does not act as a natural language question (Kolomiyets and Moens, 2011, pp.5413). Therefore, the model classifies a question as wh-interrogative only if the wh-token appears in the question.

Table 3.13 Interrogative types with examples and associated POS tags. POS tags are compliant with the Penn Treebank guidelines.

	Wh-interrogative	Polar interrogative
Interrogative tokens	Who, What, Where, Which, When, How	Is, Are, Was, Were, Do, Does, Did, Has, Have, Had
POS tags	WP, WRB, WDT	VBZ, VBP, VBD, VB
Question-1 (POS tagged)	<p> WDT NN VBZ DT <i>Which</i> <i>river</i> <i>does</i> <i>the</i> </p> <p> NNP NNP VB <i>Brooklyn Bridge</i> <i>cross?</i> </p>	<p> VBD NNP NNP VBN <i>Was</i> <i>Natalie Portman</i> <i>born</i> </p> <p> IN DT NNP NNPS <i>in</i> <i>the</i> <i>United States?</i> </p>
Answer	East River	False (No)
Answer Sentence	The Brooklyn Bridge crosses East River.	Natalie Portman was not born in the United States.
Question-2 (POS tagged)	<p> WRB JJ NNS VBD <i>How</i> <i>many</i> <i>films</i> <i>did</i> </p> <p> NNP NNP VBN <i>Hal</i> <i>Roach</i> <i>produced?</i> </p>	<p> VBZ NNP NNP <i>Is</i> <i>Christian</i> <i>Bale</i> </p> <p> VBG IN NNP <i>starring</i> <i>in</i> <i>Batman</i> </p> <p> VBZ <i>Begins?</i> </p>
Answer	509	True (Yes)
Answer Sentence	Hal Roach produced 509 films.	Christian Bale is starring in Batman Begins.

3.5.3 Pattern Extraction from Dependency Tree

Dependency Grammar (Kubler et al., 2009) introduces the concept of syntactic formation where individual tokens are linked through asymmetrical relations known as dependency relations. In essence, the dependency relation connects two tokens, one which governs the relation (head) and the other which depends (dependent). As dependency grammar expects each token of the sentence to have a head, we insert an artificial root node which actually becomes the head of the sentence to support the theoretical

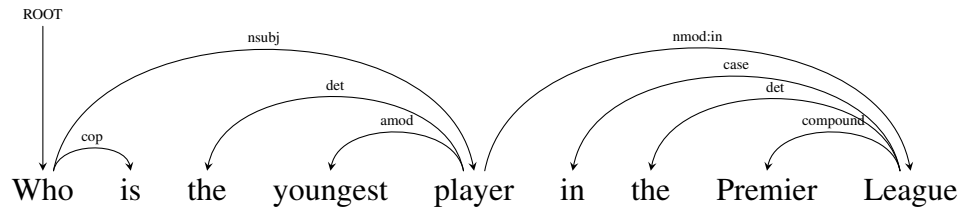


Fig. 3.7 An example depicting dependency grammar relations between tokens in a question.

and computational processing of dependency grammar. Figure 3.7 shows an example question encoded with dependency grammar relations.

Dependency parsing is the process of identifying the dependency structure of a given sentence automatically. In this research, we utilized the Stanford Parser (Manning et al., 2014), a Context Free Grammar (CFG) based parser which utilizes universal typed dependencies. The importance of adhering to universal typed dependencies will be discussed later in this section.

In our problem of dependency parsing, we denote a question Q which is composed of tokens $q_0, q_1 \dots q_n$ where q_0 is the artificially inserted root node. Consequently, $R = \{r_1, r_2, \dots, r_m\}$ is a finite set of possible dependency relation types that link two tokens in the Q . We can define the dependency tree for question Q as a directed tree T_Q where $T_Q = (V, A)$. Here, V is the spanning node set of T_Q meaning that $V \subseteq \{q_0, q_1 \dots q_n\}$ and A denotes the arcs where $A \subseteq V \times R \times V$. Also importantly, T_Q originates from the q_0 satisfying the root property which infers that there cannot be a $q_i \in V$ such that $q_i \rightarrow q_0$.

A dependency subtree in our study can be defined formally as $T_{QS} = (V_x, A_x)$ where $A_x \subseteq V_i \times R \times V_j$ and $V_x = V_i \cup V_j$. The V_i and V_j can be defined as $V_i \subseteq \{q_i | (q_0, r, q_i) \in A\}$ and $V_j \subseteq \{q_j | q_j \in V \setminus V_i\}$ respectively. This formalism limits our dependency tree to a subtree which originates from the dependent of the artificial root node.

Table 3.14 Examples of dependency subtrees extracted from parsed questions. The questions are taken from QALD-2 test dataset.

Original Typed Dependency Tree	Typed Dependency Subtree	Dependency Pattern
<p>Original typed dependency tree for the sentence "What is the official website of Tom Cruise?". The root node is labeled "ROOT". It has three outgoing arrows: one to "What" (labeled "cop"), one to "is" (labeled "nsubj"), and one to "official" (labeled "det"). "What" is connected to "is" (labeled "cop"). "is" is connected to "official" (labeled "det"). "official" is connected to "website" (labeled "amod"). "website" is connected to "of" (labeled "prep"). "of" is connected to "Tom" (labeled "pobj"). "Tom" is connected to "Cruise?" (labeled "nn").</p>	<p>Typed dependency subtree for "What is the official website of Tom Cruise?". It shows the nodes "R[wh]", "X", "X", and "X" with arrows labeled "cop" and "nsubj".</p>	$nsubj \leftrightarrow cop \leftrightarrow Root$
<p>Original typed dependency tree for the sentence "Who created Wikipedia?". The root node is labeled "ROOT". It has two outgoing arrows: one to "Who" (labeled "nsubj") and one to "created" (labeled "dobj"). "Who" is connected to "created" (labeled "nsubj"). "created" is connected to "Wikipedia?" (labeled "dobj").</p>	<p>Typed dependency subtree for "Who created Wikipedia?". It shows the nodes "X[wh]", "R", and "X" with arrows labeled "nsubj" and "dobj".</p>	$nsubj \leftrightarrow Root \leftrightarrow dobj$
<p>Original typed dependency tree for the sentence "Which river does the Brooklyn Bridge cross?". The root node is labeled "ROOT". It has four outgoing arrows: one to "Which" (labeled "det"), one to "does" (labeled "aux"), one to "the" (labeled "det"), and one to "cross?" (labeled "ROOT"). "Which" is connected to "does" (labeled "det"). "does" is connected to "the" (labeled "aux"). "the" is connected to "Brooklyn" (labeled "nm"). "Brooklyn" is connected to "Bridge" (labeled "nm"). "Bridge" is connected to "cross?" (labeled "nsubj").</p>	<p>Typed dependency subtree for "Which river does the Brooklyn Bridge cross?". It shows the nodes "X[wh]", "X", "X", "R", and "X" with arrows labeled "det", "aux", "nsbj", and "dobj".</p>	$nsubj \leftrightarrow aux + Root \leftrightarrow dobj$

Based on the aforementioned formal definition of the dependency subtree, we then extract the patterns using the subtrees identified from dependency parsing. A pattern in our approach constitutes to the dependency relations appear in the subtree. We do not place any attention on the actual tokens or their associated POS tags during the pattern extraction. This is because we only concentrate on the syntactic structure from the perspective of root and not the underlying word level features. Table 3.14 denotes an example set of dependency subtrees and patterns extracted from the original dependency trees of parsed questions. The extracted patterns represent a mere listing of relations. However, to generate a sentence utilizing these relations, the order of appearance must be declared. The final column in the Table 3.14 shows the ordered relations which can be used as the finalized pattern to generate an answer sentence.

Moreover, we use the universal typed dependencies (de Marneffe et al., 2014) in our framework. The universal typed dependencies define a taxonomy of grammatical relations which can be used across languages. This solves the key challenge in dependency parsing by allowing them to adopt for a number of languages to identify the syntactic structure. Further work on universal typed dependencies is still in progress which include mapping existing dependency schemes to this universal taxonomy (de Marneffe et al., 2014). A main reason that motivated us to employ universal typed dependencies is the opportunity to consider our current approach in a different language in the future. However, we also support dependency schemes which do not comply with universal schema. This is achieved technically by mapping universal typed dependencies to a framework specific typology for easy configuration.

The extracted patterns are preserved in a database to be utilized during the pattern search process. The next section explains the pattern searching and application process.

3.5.4 Pattern Search

When a new question and answer pair is provided to generate the answer sentence, the question is first dependency parsed and relations are extracted through the root level subtree. However, we have no prior knowledge on the ordering of the relations of the set of relations. Therefore, we now search the pattern database without considering the order of the relations and consider only the possible existence. For instance, a possible pattern $\langle nsubj, cop, dobj \rangle$ is considered as a matching pattern for the newly derived set $\langle dobj, cop, nsubj \rangle$ which is unordered. At this level of processing we have a clear idea on how the new answer sentence should be syntactically structured based on the source question, but we have no idea on the content.

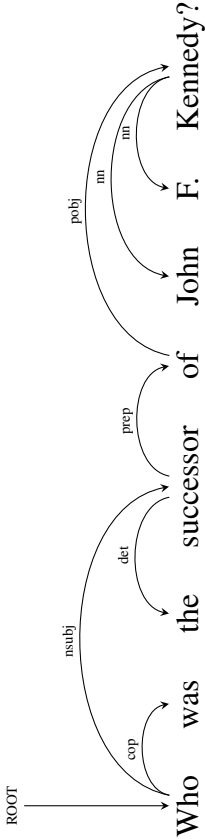
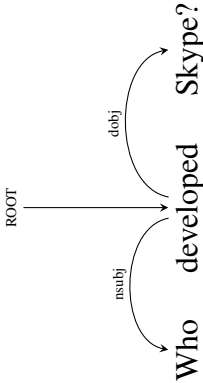
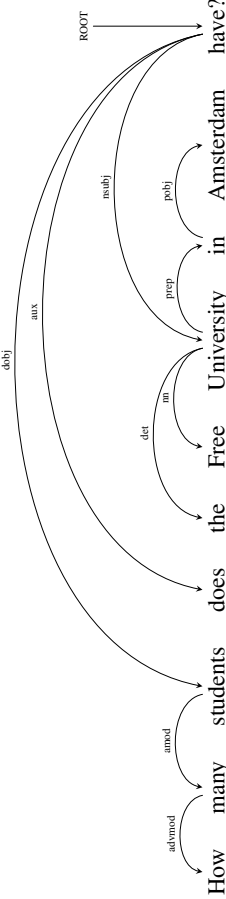
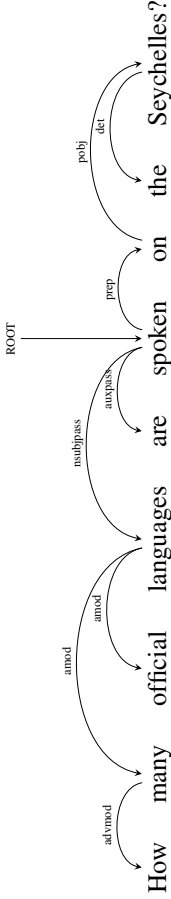
The pattern application stage looks at what content should be included in the answer sentence which will be taken from the source question. The content is derived by considering all the associated tokens in the subtrees. These tokens are now transformed into individual phrases following the same order that appear in the source question. However, we do not transform the phrase that contains the *wh*-token to a textual phrase at this stage. This is mainly to support the answer merging process which will be explained in Section 3.5.5.

Then we can order the appearance of these phrases to form an answer sentence. This is carried out by consulting the order of relations in the pattern that is selected for that particular question. Some example scenarios of phrase extraction based on the dependency tree and their ordering are shown in Table 3.15.

3.5.5 Answer Merging

The answer merging process embeds the answer to the syntactic structure of the source question. In *wh*-interrogatives, the answer merging focuses on the process of embedding another language segment to the original structure where it is appropriate. However, for

Table 3.15 Extracting and ordering phrases based on the selected pattern

Parsed Question	Selected Pattern	Extracted Phrases
 <p>Who was the successor of John F. Kennedy?</p>	$nsubj \leftrightarrow cop \leftrightarrow Root$	<ul style="list-style-type: none"> ▷ who ▷ was ▷ the successor of John F. Kennedy
 <p>Who developed Skype?</p>	$nsubj \leftrightarrow Root \leftrightarrow dobj$	<ul style="list-style-type: none"> ▷ who ▷ developed ▷ Skype
 <p>How many students does the Free University in Amsterdam have?</p>	$nsubj \leftrightarrow aux + Root \leftrightarrow dobj$	<ul style="list-style-type: none"> ▷ how many students ▷ does ▷ the Free University in Amsterdam ▷ have
 <p>How many official languages are spoken on the Seychelles?</p>	$nsubj \leftrightarrow auxpass + Root \leftrightarrow prep dobj$	<ul style="list-style-type: none"> ▷ how many official languages ▷ are ▷ spoken ▷ on the Seychelles

polar interrogatives which do not expect external answers, instead require acceptance or rejection of the provided statement in the question which in turn require modification of the polar token.

We devised a rule based model to embed the answer in wh-interrogatives based on the wh-token. The devised rules are shown in Table 3.16 for the six different wh-tokens. During the answer merging, for questions with wh-tokens except “how” will be merged with the answer utilizing the prepositions associated with that. Furthermore, certain questions need some tokens associated with the wh-token to be present in the answer sentence. For example, the example with wh-token “how” depicted in Table 3.16 needs the token “film” to be appeared in the answer sentence together with the answer. To accomplish this, we first extract the typed dependency subtree which contains the wh-token. Since the root of the typed dependency parse connects the first level dependants, the wh-token and its associated tokens are constituted to a single subtree in the complete parse tree. The extracted subtree with the wh-token is then analysed to check whether there are prepositions associated with that to generate the phrase by merging the answer.

Another important factor is that the specified wh-tokens do not include “why”. This is due to two reasons; firstly the current research is only dealing with factoid questions and “why” is the wh-token associated with the definitional questions. Secondly, definitional questions do not need answer sentences, as the answer is a textual definition which do not need to improve for further processing targeting presentation.

In addition to placing the factoid answer in the linguistic structure of the source question, the module also embeds measurement units and converts the numbers to verbal representation where necessary. For example, if a number appears at the start of the resulting answer sentence, then the number is converted to verbal form (e.g., 24 \Rightarrow Twenty four) otherwise it is left as it is. Additionally, some questions require measured values as answers (e.g., height, weight). These answers are associated with the

Table 3.16 Rules applied in the answer merging for wh-interrogatives

Wh-token	Merging scheme	Example phrases	Merged answer example
Which	Existing preposition + Answer	in which country	in New Zealand
What	Existing preposition + Answer	for what city	for London
Whom	Existing preposition + Answer	for whom	for Barack Obama
How	Verbalized answer (once/twice/thrice)	how often	twice
	Factoid answer + Rest of the phrase	how many films	509 films
When	Existing preposition + Answer New preposition (based on answer) + Answer	from when when	from 1990 in 1990
Where	New preposition + Answer	where	in Auckland

measurement unit during the answer merging phase with the help of the measurement unit information database introduced in Section 3.2.3.2.

To identify the measurement unit we exploit the SPARQL query. The framework first parses the SPARQL query and extracts the basic graph patterns and embedded triples mentioned in the query. Listing 3.5 and Listing 3.6 show a SPARQL query and a resulting SPARQL algebraic definition for basic graph patterns. Although the example depicts a scenario with one triple, a more complex SPARQL query can result in multiple basic graph patterns each having multiple triples. In the next step we screen the triples to find out the triple which contains the queried variable (e.g., *?num* in the example shown in Listing 3.5). The predicate in this triple is the queried predicate and the module searches the measurement unit information database (explained in Section 3.2.3.2) to find the measurement unit associated with the predicate.

```

1 PREFIX dbo: <http://dbpedia.org/ontology/>
2 PREFIX res: <http://dbpedia.org/resource/>
3 SELECT ?num WHERE
4 {
5   res:Michael_Jordan dbo:height ?num .
6 }

```

Listing 3.5 SPARQL query for the question “How tall is Michael Jordan?”

```

1 (project (?num)
2  (bgp
3   (triple <http://dbpedia.org/resource/Michael_Jordan> <http://dbpedia.org/ontology/height> ?num)))

```

Listing 3.6 SPARQL algebraic expression of the query shown in Listing 3.5

3.5.6 Sentence Realization

The sentence realization can be thought of as a linguistic realization (in other words this is a revision to the already built sentence) on the already generated sentence to make it more natural. However, by this stage, the answer sentence is nearly built except the verb inflections. Therefore, this module focuses on realization of periphrastic tense in occasions where the verb can be inflected without comprising the semantics (e.g., does cross \Rightarrow crosses). To inflect the verbs where necessary, the verb information database discussed in Section 3.3.1 is utilized. Table 3.17 shows four examples of the answer sentence generation using the typed dependency subtree patterns and the realization of the generated answer sentence to further naturalize it.

3.6 Lexicalization

Lexicalization is the first step for generating an entity description which will be shown to the user with an answer sentence generated in the previous section. Throughout the

Table 3.17 Examples of answer sentence generation with realization. Note that tokens highlighted in yellow are the periphrastic tense which are realized in the final answer sentence and tokens highlighted in blue represent the embedded answer. Note that the initial answer sentence is a manipulation of a dependency parsed question and not a dependency parsed sentence.

Dependency Parsed Question	Pattern	Initial answer sentence	Realized answer sentence
<p>How many children did Benjamin Franklin have?</p>	nsubj ↔ aux+Root ↔ dobj[wh]	<p>Benjamin Franklin had 3 children</p>	Benjamin Franklin had 3 children
<p>When did Michael Jackson die?</p>	nsubj ↔ aux+Root ↔ advmod [wh]	<p>Michael Jackson died on June 24, 2009</p>	Michael Jackson died on June 24, 2009
<p>How many employees does Google have?</p>	nsubj ↔ aux+Root ↔ dobj [wh]	<p>Google has 49829 employees</p>	Google has 49829 employees
<p>How often did Nicole Kidman marry?</p>	nsubj ↔ aux+Root ↔ advmod [wh]	<p>Nicole Kidman married thrice</p>	Nicole Kidman married thrice

NLP literature, lexicalization is one of the terms that has many different definitions. Therefore, before diving into the technical details we first define the lexicalization in the context of this research.

In this research, lexicalization counts for two things; to generate patterns that transform triples to natural language sentences and secondly a method of searching a matching pattern for a given triple and further realizing it through a revision process. The pattern generation process is an ensemble of four different pattern processing modules which focus on different aspects of deriving patterns. The realization and revision process transforms an existing matching pattern to suit with the new triple. Furthermore, the lexicalization patterns that we build in this process are targeting individual triples. This approach is more difficult than converting an existing triple graph to natural language using mapping triple elements to existing text. However, individual lexicalization of triples is widely usable as it can be used with any given triple collection. These features make our lexicalization approach significant from other existing methods in verbalization and lexicalization as described in previous research such as Walter et al. (2013), Ell and Harth (2014), and Duma and Klein (2013).

Ell and Harth (2014), and Duma and Klein (2013) focused on triple graph lexicalization which lexicalizes individual triples which is different to our approach. In comparison, Walter et al. (2013) employed dependency parsing to extract patterns to lexicalize triples which more closely resembles with our strategy. This approach first identifies the sentences that contain the triple subject and object, followed by a dependency parse of the sentences. The model then looks for the shortest dependency path between the subject and object and extracts it if it exists. The substring that appears in the shortest dependency path becomes the pattern to lexicalize the triple. Although the model is promising when considering it from a linguistic viewpoint, there exist a number of challenges. Firstly, the sentence collection is used without any preprocessing which

Table 3.18 Example lexicalization patterns

Triple	Lexicalization Pattern
$\langle \text{Steve Jobs, founder, Apple Inc} \rangle_T$	$\langle S?, \text{ is the founder of, } O? \rangle_L$
$\langle \text{Klaus Wowereit, party, Social Democratic Party} \rangle_T$	$\langle S?, \text{ is a member of, } O? \rangle_L$
$\langle \text{Canada, currency, Canadian dollar} \rangle_T$	$\langle O?, \text{ is the official currency of, } S? \rangle_L$
$\langle \text{Canada, capital, Ottawa} \rangle_T$	$\langle O?, \text{ is the capital city of, } S? \rangle_L$
$\langle \text{Rick Perry, birthDate, 1950-03-04} \rangle_T$	$\langle S?, \text{ was born on, } O? \rangle_L$

compromises the scope of the collection since preprocessing tasks such as co-reference resolution can greatly increase the scope for extracting a pattern by searching the subject and object of the triple which include the co-referents as well as the antecedents. Furthermore, the dependency paths are naively extracted and introduced as patterns without any post-processing. The main drawback here is that adjectives and adverbs are also taken as a part of the pattern which limits the generalizability of the pattern. Apart from the aforementioned drawbacks, Walter et al. (2013) do not consider the conditions under which the acquired patterns should be applied on new triples. For instance, the gender of the triple subject, and ontology class of the subject can determine whether a pattern is suitable or not. This model seems to have very limited applications due to the absence of such metadata.

Table 3.18 shows some sample lexicalization patterns that we target to build from the lexicalization process. As shown in the table, lexicalization module simple does not look for a lexical choice, however, it is a syntactically correct, semantically appropriate pattern which can transform the triple to a basic natural language segment.

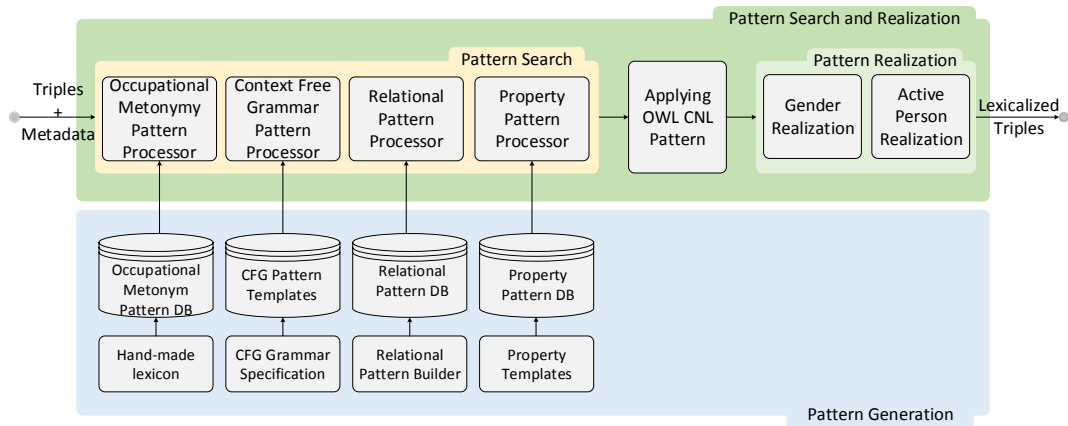


Fig. 3.8 Lexicalization high-level architecture

We first explain the lexicalization architecture in Section 3.6.1. The architecture described here is an ensemble of four modules which will be discussed thereafter in Section 3.6.2, 3.6.3, 3.6.4, and 3.6.5.

3.6.1 Lexicalization High-level Architecture

Lexicalization architecture which we introduce here is an ensemble revision architecture. Technically, the architecture is composed of four modules which will be executed in a sequence to generate lexicalization patterns. In addition, these patterns are further realized using the revision processes implemented.

Figure 3.8 depicts the high-level overview of the lexicalization architecture. The complete lexicalization can be explained in two forms; lexicalization pattern generation and pattern search.

The pattern generation process utilizes the occupational metonym patterns, context free grammar patterns, relational patterns, and property patterns to generate lexicalization patterns. The following sections explain these different pattern types and discuss how different patterns are generated.

As shown in Fig. 3.8, the lexicalization pattern search module seeks for patterns in a sequence. The prioritization of the lexicalization modules in the implementation is the same order as depicted in the figure. The occupational metonym patterns are placed first as it is a limited lexicon which is built through human supervision and we have done several rounds of validation to make sure that the patterns are valid and applicable. The process of building the occupational metonym pattern database and its linguistic background of the occupational metonym formation can be found in Section 3.6.2. The Context Free Grammar (CFG) (Bundy and Wallen, 1984) is generally declared as a theory of language generation as well as language understanding. A limited set of CFG rules are used in the module which are already validated. Therefore, this module is placed next to the metonym patterns. The relation patterns are extracted from free text and therefore placed after aforementioned two modules. However, the property patterns are again a lexicon which is built under supervision. The reason to place property patterns next to relation patterns is that the wide applicability of property patterns can make relational patterns hidden. The issue arises in such a scenario is that the generated language output tends to be in a unique language style which loses the linguistic variety. Due to this reason the relational patterns are considered before the property patterns.

Once the pattern processing modules apply the most suitable lexicalization pattern for a triple, we generate a pattern based on the core ontology class of the triples. Since pattern search module works on triples from one entity at a time, there can be only one core ontology class based pattern (every entity belongs to exactly one ontology class hierarchy). Section 3.6.6.2 describes the process of applying an ontology class based Controlled Natural Language (CNL) pattern.

Once the triples are associated with lexicalization patterns, we apply revision steps to further realize the applied patterns. This process contains two modules; gender

realization and person active realization. Section 3.6.6.3 and 3.6.6.4 describe the two realization processes in detail.

3.6.2 Occupational Metonym Patterns

Metonym is a single word or phrase which is referred not by its own name, but by a name that is associated with the meaning of it. A well understood and highly used metonym is “Hollywood”, which is used to denote the USA film industry. In the same way, there exist several metonyms which are created based on the occupations which are introduced as occupational metonyms (Alexiadou and Schäfer, 2008, 2010; Panther and Thornburg, 2002). Some of them are “commander”, “owner”, and “producer” which are used respectively to denote someone who gives commands to one or more people, someone who owns something, someone who produced something.

3.6.2.1 Morphological Formation

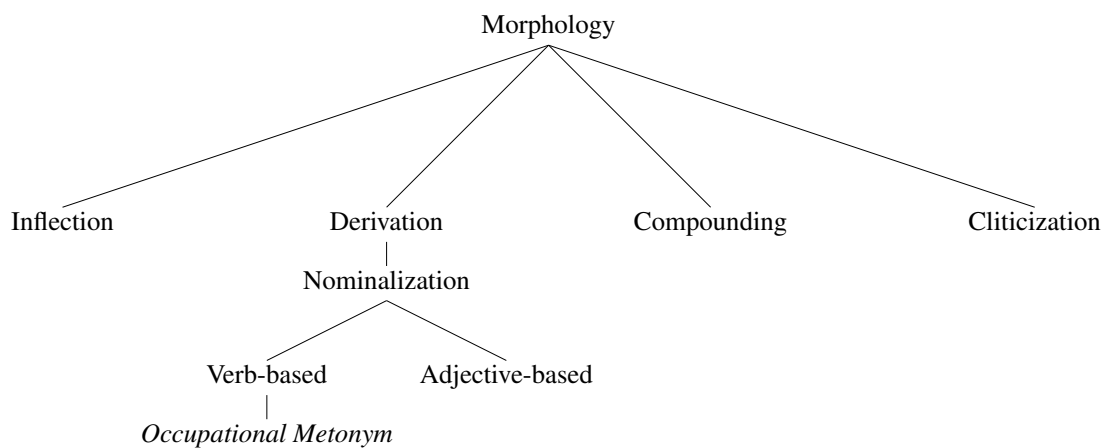


Fig. 3.9 Classification hierarchy of English morphology

Figure 3.9 shows the classification hierarchy of English morphology and highlights under which category occupational metonyms are classified. Based on this classification, it is clear that occupational metonyms are a nominalization of verbs.

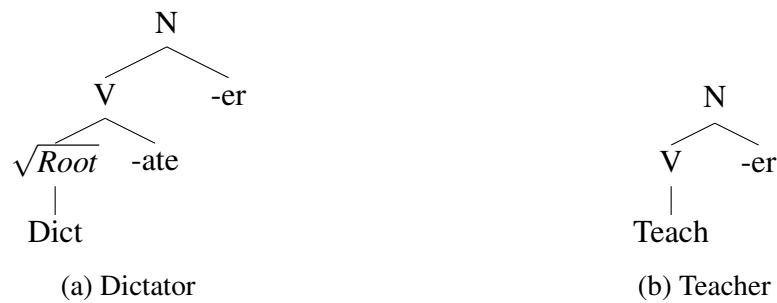


Fig. 3.10 Two different occupational metonym formation applying -er nominals

Two widely used forms of nominalization for occupation metonyms is the affixing of so-called agentive nominals; *-er* and *-or* nominals. These nominalizations can be directly applied on a base verb as well as on top of other morphological inflections. For example, Fig. 3.10a and Fig. 3.10b show two different occupational metonym forms in different granularity of applying nominalizations to form occupational metonyms.

Although it is possible to develop an unsupervised lexicon by nominalizing verbs, the idiosyncrasy of English makes it rather difficult. In some cases, the nominalized noun may also refer to non-agentive nominals (Schäfer, 2011).

- scratcher - a scratched lottery ticket
- broiler - a broiled chicken

There are multiple occasions where aforementioned occupational metonyms appear as predicates of the triple. For example, the triple $\langle \textit{Batman Begins}, \textit{publisher}, \textit{Nolan} \rangle_T$ contains the “publisher” as the predicate which is an *-er* nominalized form of the verb “publish”. Since the base verb of the nominalization indicates the verb related to the profession, we can specify a straightforward lexicalization as “*Christopher Nolan published Batman Begins*”. However, a further realization of the pattern can be formed by a triple subject prominent lexicalized version as “*Batman Begins is published by Christopher Nolan*”.

We utilize the occupational metonyms to build a lexicon which associates triples with lexicalization patterns. In essence, if a triple has an occupational metonym as the predicate, then that triple is associated with the corresponding lexicalization pattern from the lexicon. For instance, consider the triple $\langle \textit{Batman Begins}, \textit{publisher}, \textit{Nolan} \rangle_T$ which has the occupational metonym “*publisher*” as the predicate. This triple is lexicalized with a lexicalization pattern such as $\langle S?, \textit{was published by}, O? \rangle_L$ which is based on the nominalized verb of the publisher. In addition to the predicate and the lexicalization pattern, the lexicon also records the ontology class that the pattern can be applied for. This is for two reasons; firstly it supports the RDF inference during the pattern search which is discussed in detail in Section 3.6.6.1. Secondly, the same predicate may appear under different ontology class hierarchies, however, the lexicalization may depend on the core ontology class. For instance, consider the predicate “*author*” which appears under two ontology class hierarchies: $Work \rightarrow \textit{Art Work}$ and $Work \rightarrow \textit{Software}$. The lexicalization patterns for the same predicate under these two ontology class hierarchies will be as $\langle S?, \textit{was painted by}, O? \rangle_L$ and $\langle S?, \textit{was developed by}, O? \rangle_L$. The difference in the lexicalization is caused by the ontology class hierarchy to which the predicate belongs.

3.6.2.2 Generalizing Occupational Metonyms

In addition to building the occupational metonym lexicon, a further step is taken to associate the metonyms with synonymous predicates. This is to further extend the metonym lexicon and make it widely applicable. This step is taken because DBpedia contains synonyms for occupational metonyms and in which case cannot be associated with a pattern. Although, revising these synonyms to well-defined ontology properties is an essential task in the Linked Data domain, however, this is not yet accomplished

and remains a future goal. Therefore, the generalization property provides a solution to associate such ontology properties with an occupational metonym.

3.6.3 Context Free Grammar Patterns

Context Free Grammar (CFG) (Bundy and Wallen, 1984) is considered dual purpose in NLP. This means that it can be used to understand the language as well as to generate language based on given grammar rules. For instance, Busemann (2005) describes the TG/2 where CFG rules are associated with templates to provide natural language text. Recently, Stribling et al. (2005) demonstrated the SCiGen program which generates scientific papers using handwritten CFG rules. However, a burden associated with CFG is that the grammar rules need to be specified in advance, either as handwritten rules or as phrase structure trees derived from a seed corpus.

Due to these burdens associated with CFG based language production, our system does not use CFG as a main source. Only certain predicates which satisfy a predetermined constraint are associated with CFG pattern. The constraint is that the predicate must either be a verb in past tense (e.g., influenced) or a predicate that is provided in passive form (e.g., maintained by). The CFG basic grammar form (\mathcal{G}) for single sentence level construction can be illustrated as below.

$$S \rightarrow NP VP$$

$$NP \rightarrow NNP$$

$$VP \rightarrow VBD NP$$

where S denotes a sentence and NP , NNP , VP , and VBD represent a noun phrase, proper noun, verb phrase, and verb in past tense respectively.

The CFG patterns are applied to the triples with predicates which are identified as verbs in past tense and if the identified verb has a frame $NP \leftrightarrow VP \leftrightarrow NP$ (identified using verb information database discussed in Section 3.3.1). For an example, the triple

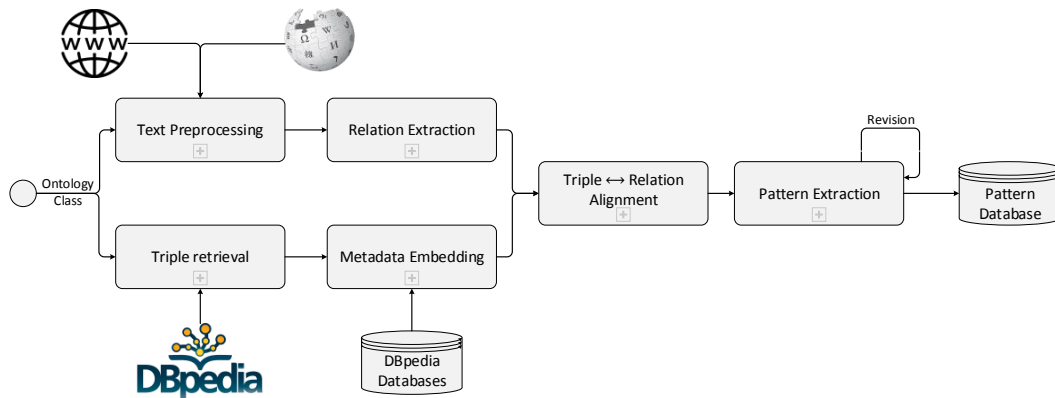


Fig. 3.11 Relational pattern extraction process

$\langle \textit{Socrates}, \textit{influenced}, \textit{Plato} \rangle_T$ can be lexicalized as its predicate satisfies the above CFG rule (i.e., $NP \leftrightarrow VP \leftrightarrow NP$; in essence the verb “*influence*” has the required frame. In addition to these types of triples, CFG pattern processing module also covers the predicates which are passive form verbs (e.g., $\langle \textit{Aristotle}, \textit{influencedBy}, \textit{Parmenides} \rangle_T \Rightarrow \langle \textit{Aristotle}, \textit{was influenced by}, \textit{Parmenides} \rangle_{LT}$).

3.6.4 Relational Patterns

Relational patterns are lexicalization patterns which are derived from the unstructured text using relation extraction. In brief, we process a large number of unstructured text resources related to the triples and extract relations using Open Information Extraction (OpenIE). These relations are then aligned with the triples to extract lexicalization patterns. Figure 3.11 depicts the schematic representation of the relational pattern extraction process.

The module is initiated with an ontology class hierarchy and associated entity collection. Table 3.19 depicts a sample input to the framework.

The module takes the aforementioned input and then moves to a parallel process of relation extraction and triple retrieval. During this process, it collects text related

Table 3.19 Sample input to the relational pattern extraction module. The example shows two ontology class hierarchies and associated entities. The actual input contains a series of class hierarchies and their associated entities.

	Agent → Person	Agent → Organisation → Company
Entities	Jimmy Wales	Google
	Larry Sanger	Intel
	Natalie Portman	Microsoft

to each of the entities provided and then extracts relations from the collected text. In parallel, triples related to these entities are retrieved from the DBpedia and enriched with metadata which is needed for the latter processes. The relations are then aligned with triples to extract relational patterns.

We also record some meta data for each relation pattern being processed. The metadata include ontology class hierarchy, natural gender of the subject, object multiplicity, and two more factors – occurrence count and relation confidence which are discussed in Section 3.6.4.5. The main objective of recording these metadata is that our preliminary analysis has shown these can place restrictions when applying lexicalization patterns.

Ontology Class Hierarchy The lexicalization patterns that are extracted for triples can be specific to the ontology class that they belong to. For instance, consider two triples, $\langle \textit{Skype}, \textit{author}, \textit{Janus Friis} \rangle_T$ and $\langle \textit{The Scream}, \textit{author}, \textit{Edvard Munch} \rangle_T$, which are retrieved from DBpedia. Both triples contain the same predicate “*author*”, however, the entities described here belong to two different ontology classes, “*Software*” and “*Art Work*” respectively. The first triple can be lexicalized as “*Skype is developed by Janus Friis*”, meanwhile the second triple will be generally lexicalized as “*The Scream is painted by Edvard Munch*”. This differentiation is caused by the fine ontology class that the subjects of the two entities belong to. This emphasises that associating the ontology hierarchy with

the lexicalization pattern is important when searching a matching pattern for a new triple.

Gender Gender of a subject is another property that affects the lexicalization pattern not generalizable across all entities that are associated with a particular predicate. For instance consider the two triples, $\langle \textit{Barack Obama}, \textit{spouse}, \textit{Michelle Obama} \rangle_T$ and $\langle \textit{Michelle Obama}, \textit{spouse}, \textit{Barack Obama} \rangle_T$. Although they have the same predicate and both subjects belong to the same fine ontology class, a lexicalization pattern generated for the first triples such as $\langle S?, \textit{is the husband of}, O? \rangle_L$ cannot be used for the second triple as the gender of subjects are different. Due to this fact the framework also associates the gender of the subject with the retrieved triple. We consult the database described in Section 3.2.3.1 to find the gender of a subject .

Object Multiplicity Some triples contain the same subject and predicate with different objects. These triples with multiple objects require different natural language representation compared to another predicate with a single object. For example consider triples related to *Nile River*, $\langle \textit{Nile}, \textit{country}, \textit{Egypt} \rangle_T$, $\langle \textit{Nile}, \textit{country}, \textit{Rwanda} \rangle_T$, and $\langle \textit{Nile}, \textit{country}, \textit{Uganda} \rangle_T$ which describe the countries that the Nile River flows through. However, the same information is represented for *East River* as $\langle \textit{East River}, \textit{country}, \textit{USA} \rangle_T$ which describes that *East River* is located in USA. These two scenarios need two different lexicalization patterns such as $\langle S?, \textit{flows through}, O? \rangle_L$ and $\langle S?, \textit{located in}, O? \rangle_L$ respectively. This shows that object multiplicity plays a crucial role in deciding the most appropriate lexicalization pattern for a given triple.

The below sections describe the pattern extraction process in detail which uses a unstructured text resource to extract lexicalization patterns.

3.6.4.1 Text Preprocessing

We first retrieve unstructured text related to the entities from Wikipedia as well as from web based text resources. Since DBpedia contains information extracted from Wikipedia (i.e. Wikipedia Infoboxes which contain structured data are converted to Linked Data), it is considered as a primary resource for text extracted. Articles extracted from Wikipedia are wrapped in an HTML boilerplate and this causes a serious issue when extracting pure text representation of the article. To address this the module employs the Boilerpipe (Kohlschütter et al., 2010), a shallow text feature based boilerplate removal algorithm.

However, Wikipedia itself is not sufficient to build a text corpus to extract a wide range of relations. Therefore, we extract text from other web resources when building the text corpus.

What we expect from this text is a description related to a particular entity. Also sentences in the description should discuss information related to the entity. However, the text extracted from this step can contain co-references to already mentioned entities. Such conferences cannot be resolved once the relation extraction is performed. Therefore, as a preprocessing task we resolve the co-references by applying the entity full name. For example a paragraph like,

“*Abraham Lincoln* is regarded as one of America’s greatest heroes. *He* is a remarkable story of the rise from humble beginnings to achieve the highest office in the land.”

will be converted to,

“*Abraham Lincoln* is regarded as one of America’s greatest heroes. *Abraham Lincoln* is a remarkable story of the rise from humble beginnings to achieve the highest office in the land.”

We utilized the Stanford CoreNLP (Lee et al., 2011; Manning et al., 2014) to find co-references and map them to the antecedent. Then the mapped co-references are

substituted with the antecedent text. This process ensures that extracted relations do not contain anaphors, as existence of such makes it difficult to align relations with corresponding triples. The result of this process, co-reference resolved set of sentences, is passed to the relation extraction process.

3.6.4.2 Triple Retrieval

The triples are retrieved from RDF/XML files associated with each entity that is identified in the SPARQL query. Apache Jena is employed to parse the RDF/XML file and extract triples related to the entity. Furthermore, as stressed in Section 3.2, only triples under the DBpedia ontology schema are extracted for the lexicalization process. We have also identified some triples which are not useful in the lexicalization process. These triples include Wikipedia links, image links, and other identifiers (e.g., VIAF ID). These triples are removed from the triple collection.

3.6.4.3 Relation Extraction

The task of relation extraction is to extract relation triples from the co-reference resolved text. The approaches towards relation extraction can be categorized into two camps; Closed Information Extraction (ClosedIE) and Open Information Extraction (OpenIE) (Etzioni et al., 2008).

The ClosedIE which is the traditional approach towards the relation extraction attempts to extract natural language relations between two mentioned entities. This approach relies on rule based methods, kernel methods and sequence labelling methods. These methods brings several key drawbacks to ClosedIE such as the need for hand-crafted rules, need for hand-tagged data, and difficulties in domain adaptability.

However, when applying relation extraction in this project, we focused on a domain independent method, which looks at linguistic structure of the sentence and extracts relations.

The recently proposed OpenIE, promised to work in a large scale corpus such as web (web as a corpus). The OpenIE approach for relation extraction deviates from the traditional relation extraction process significantly. OpenIE identifies relations using relational phrases. A relational phrase is a natural language phrase that denotes a relation in a particular language. The identification of such relational phrases makes the system scalable by extracting an arbitrary number of relations without tagged data. Furthermore, as relational phrases are based on linguistic knowledge and do not involve domain knowledge, OpenIE can work in multiple domains without training instances.

We used Ollie (Mausam et al., 2012) - OpenIE system in this module. Ollie has several advantages over the other two analysed systems, ClauseIE (Del Corro and Gemulla, 2013) and Reverb (Fader et al., 2011). ClauseIE is a clause based OpenIE module which performs on a pre-specified set of clauses derived from dependency parsing. Due to this specification, ClauseIE is unable to find many linguistic structures outside its scope. As Ollie is trained on a large number of instances, it can extract several relations which are not covered by ClauseIE. On the other hand, Ollie is the successor of Reverb, and hence Ollie has significant improvements over Reverb.

3.6.4.4 Triple-Relation Alignment

Once the relations are extracted using the OpenIE, we then align each relation with the triple to identify candidate relations which can be considered as lexicalization patterns. The aligner is mainly focused on mapping the subject and object of a triple with the arguments of a relation. To accomplish this mapping we employ the word overlapping measure. In particular, we employ the Phrasal Overlap Measure (POM) which is the

best performer in word overlap category based on the extensive empirical research carried out by Achananuparp et al. (2008). Furthermore, POM is capable of favouring the occurrence of a complete phrase, which is an essential feature in triple-relation alignment. Measurement of POM is calculated according to equation (3.1).

$$sim_{overlap,phrase}(T_c, R_c) = \tanh\left(\frac{overlap_{phrase}(T_c, R_c)}{|T_c| + |R_c|}\right) \quad (3.1)$$

where, T_c and R_c represent triple components (i.e., subject or object) and relation components (i.e., arg_1 or arg_2 of the relation) respectively, and $overlap_{phrase}(T_c, R_c)$ is calculated using equation (3.2) for m phrasal n -word overlaps.

$$overlap_{phrase}(T_c, R_c) = \sum_{i=1}^n \sum_m i^2 \quad (3.2)$$

where, m is a number of i -word phrases that appear in text string pairs.

Since we are not aware of the order the relation and triple should be mapped (e.g., $arg_1 \Rightarrow$ subject or $arg_1 \Rightarrow$ object), we first calculate the POM for each combination. We use the decision process depicted in Table 3.20 to determine the final alignment. According to the table, a higher alignment determines the triple component that should align with the relation component. Once the alignment is decided and phrasal overlap measure is calculated, we multiply subject and object alignments to get the final alignment between the complete triple and the relation.

The overlapping of text is calculated based on the exact textual representation. However, there can be scenarios where the object of a triple has more than one representation. For example, a date can be represented in multiple formats in natural language. Therefore, when calculating the overlap between the triple object and the relational argument phrase, all possible formats and verbalizations of the triple object must be

Table 3.20 Decision process for relation-triple alignment. This is used to determine how subject and object from triple should align with two arguments from the relation.

Condition	Alignment
$(sim_{pom}(subject, arg_1) > sim_{pom}(subject, arg_2)) \wedge$ $(sim_{pom}(object, arg_2) > sim_{pom}(object, arg_1))$	$(subject \xrightarrow{align} arg_1) \wedge$ $(object \xrightarrow{align} arg_2)$
$(sim_{pom}(subject, arg_2) > sim_{pom}(subject, arg_1)) \wedge$ $(sim_{pom}(object, arg_1) > sim_{pom}(object, arg_2))$	$(subject \xrightarrow{align} arg_2) \wedge$ $(object \xrightarrow{align} arg_1)$

consulted. The list below shows the verbalizations carried out to support phrasal overlap matching.

Date The dates are verbalized for phrase matching by converting the date form to 7 different formats.

Measured Values Triple objects which are measured values can be represented in multiple ways by associating them with different measurement units. However, the challenge is that DBpedia does not provide the measurement unit of the original triple object value. To overcome this, a database is created which maps triple objects (only measured ones) to the measurement units.

Normal Numbers Normal numbers are transformed to different scales as well as to verbal representation.

3.6.4.5 Pattern Extraction

The pattern extraction process elicits a lexicalization pattern from the aligned relation by substituting them with expressions. In essence we represent the subject as $S?$ and object as $O?$. A naive replacement of subject and object as illustrated in the example shown in Fig. 3.12 cannot be accomplished here due to the following two reasons explained below:

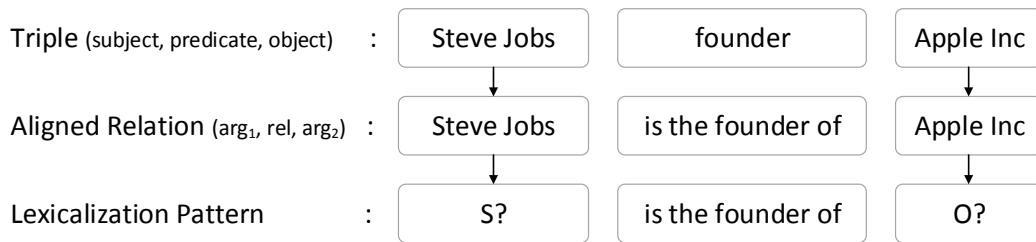


Fig. 3.12 Extracting a lexicalization pattern by matching the aligned relation to a triple. The scenario depicts an exact match where triple subject and object are exactly matched to the two arguments of the relation.

- Relation arguments are mapped with one of the verbalizations instead of the triple object.

If the relation object is aligned with one of the verbalizations of the object value, then direct replacement can cause information loss of unnecessary information being included in the pattern. To avoid this, the module searches for each verbalization in the triple argument and then replaces them with the required token.

- Triple subject or object can be mapped with the compound token from the relation argument.

Consider the below example where a triple and an argument are provided which has an acceptable alignment score.

Triple: $\langle \textit{Steve Jobs}, \textit{spouse}, \textit{Laurene Jobs} \rangle_T$

Relation: $\langle \textit{Steve Jobs}, \textit{was married to}, \textit{Laurene Powell Jobs} \rangle_R$

In the above scenario, the triple object is mapped to the relation arg₂ which is expressive. A partial substitution of the triple object is possible in such scenarios, however, they result in inaccurate data by leaving some tokens unaffected. To solve this issue we introduce the dependency tree based compound token substitution. We first aggregate the relation, so that it is transferred to a natural language sentence. This sentence is then dependency parsed and typed dependencies are extracted for the relation

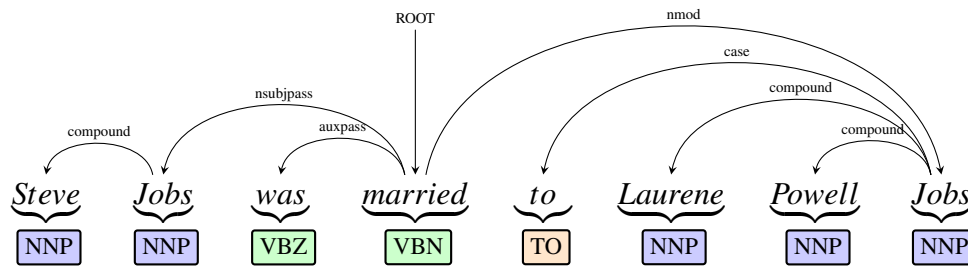


Fig. 3.13 Typed dependency parse to identify compound tokens using compound typed dependency relationship

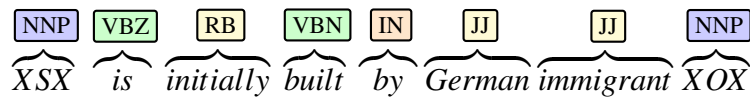


Fig. 3.14 POS tagged transformed sentence

argument. Figure 3.13 shows the typed dependency parse of the aforementioned relation where “*Laurene Powell Jobs*” appears in a compound relation. Typed dependencies which represent the compound relations are transformed back to multi-word phrases and other tokens are kept as separate. Next each multi-word phrase is checked whether it contains the triple object tokens in full. In the occasion of such scenario, the multi-word phrase is substituted with the triple object value.

In addition to the above, a post-processing step is designed to extract cohesive patterns which can be generalized regardless of the entity it is associated with. This cohesion is focused on filtering adjectives and adverbs from the text. The extracted pattern is first transformed to a natural language sentence by aggregating them and replacing subject and object expressions ($S?$ and $O?$) with proper nouns (XSX and XOX) to avoid parser misclassification by taking the punctuations of the expressions into account. Figure 3.14 depicts an example scenario where presence of adjectives makes the patterns specific to a single entity. The example pattern is extracted from the sentence “Brooklyn Bridge is initially built by German immigrant John A. Roebling” for the triple $\langle Brooklyn Bridge, architect, John A. Roebling \rangle_T$.

In addition to the aforementioned tasks, relational pattern extraction needs a threshold point to select a lexicalization pattern. This is because relational patterns come with different alignment scores. In the research we set this value to 0.21 as this value is corresponding to the single token matching in the alignment. In essence, we consider the case where subject and object are composed of one token and match with the relation components which results in 0.21.

The pattern extraction process also records the following information which will later helps us to select the best relational lexicalization pattern in the search phase.

Occurrence count Occurrence count of a pattern measures how many times the same lexicalization pattern is extracted from the text for a certain predicate but for different entities. The main objective of the consulting occurrence count is that if a pattern is appearing repeatedly for the same predicate but different entities, then that pattern can be considered as a generic pattern which can be used to lexicalize the predicate. For instance consider the scenario that, *almaMater* is a predicate for *Person* ontology class the two entities *Natalie Portman* and *Anna Koranikowa*. The relations that are derived for both entities are shown in Table 3.21 with their respective alignment scores and occurrence counts.

All lexicalization patterns in Table 3.21 are derived with the same alignment score. However, the pattern $\langle S?, \textit{graduated from}, O? \rangle_L$ has appeared twice while $\langle S?, \textit{cheated at}, O? \rangle_L$ has appeared only once. The latter is of course only specific to *Natalie Portman* and cannot be used to convey the semantic associated with the predicate *almaMater*. Therefore, the occurrence count has to act as a factor to determine whether the lexicalization pattern is specific only to one entity and convey the required semantic associated with the corresponding predicate.

Relation confidence score The OpenIE relation extraction process which is explained in Section 3.6.4.3 generates a confidence score for each relation. More informa-

Table 3.21 Lexicalization pattern with their respective scores and occurrence counts.

Natalie Portman	Anna Koranikowa	
Relations and confidence scores		
$\langle \textit{Natalie Portman, graduated from, Harvard University} \rangle_R$	0.7345	$\langle \textit{Anna Koranikowa, graduated from, Oxford University} \rangle_R$
$\langle \textit{Natalie Portman, cheated at, Harvard University} \rangle_R$	0.7345	
Lexicalization patterns, and alignment scores, and occurrence counts		
$\langle S?, graduated from, O? \rangle_L$	0.8192	2
$\langle S?, cheated at, O? \rangle_L$	0.8192	1

tion on the confidence function can be found in Mausam et al. (2012). With each lexicalization pattern we also report the confidence score of the relation which is used to derive the lexicalization pattern. The importance of the confidence score in the lexicalization pattern search phase will be discussed in Section 3.6.6.1.

3.6.5 Property Patterns

Property patterns specify a limited lexicon where certain predicates are associated with pre-specified list of templates as lexicalization patterns. Five such patterns are specified which will be applied only to the predetermined list of predicates. Table 3.22 lists the five patterns with examples of lexicalization when applied to triples with predetermined predicates. As shown in the Table 3.22, the pattern contains three triple expressions which will be replaced by their verbalized form during the lexicalization. The module is designed in such a way that it can be scaled with newly defined property patterns without additional effort.

Since the property pattern module is at the end of the pattern processing sequence, some of the triples may still use the pattern determined in a previous module instead

Table 3.22 Property patterns with examples

ID	Pattern	Predicate	Triple	Resulting lexicalization
PP-1	$\langle S?'s P?, is, O?\rangle_L$	height	$\langle Michale\ Jordan, height, 1.98\rangle_T$	$\langle Michael\ Jordan's\ height, is, 1.98\rangle_{LT}$
PP-2	$\langle S?, has, O? P?\rangle_L$	championships	$\langle Rubens\ Bar-richello, wins, 0\rangle_T$	$\langle Rubens\ Bar-richello, has, 0\ championships\rangle_{LT}$
PP-3	$\langle S?, is, O?\rangle_L$	occupation	$\langle Natalie\ Portman, occupation, an\ actress\rangle_T$	$\langle Natalie\ Portman, is, an\ actress\rangle_{LT}$
PP-4	$\langle P? in S?, is, O?\rangle_L$	largestCity	$\langle Australia, largesCity, Sydney\rangle_T$	$\langle Largest\ City\ in\ Australia, is, Sydney\rangle_{LT}$
PP-5	$\langle S?, P?, O?\rangle_L$	isPartOf	$\langle Delft, isPartOf, South\ Holland\rangle_T$	$\langle Delft, is\ part\ of, South\ Holland\rangle_{LT}$

of the property pattern thus causing the property pattern to be ignored. This setting is arranged if majority of the triples are lexicalized with the property patterns, then the linguistic variation is negatively affected by having more similar sentences throughout a passage. Since language variety is one of the factors that make language naturalize, the framework attempts to maintain the variety to a level that it can achieve with the current settings.

Another important factor to notice in property patterns is that they are not associated with the ontology class of the subject. This is intentionally left in order to generalize the property patterns and apply them in a wide scale thus providing at least a basic lexicalization for majority of the triples.

3.6.6 Pattern Search and Realization

This section describes the process of searching for a pattern, applying the selected pattern to triples, and further realizations. We start the discussion in Section 3.6.6.1 by explaining the process of searching and applying patterns which are introduced in previous sections. Section 3.6.6.2 describes the method of utilizing an ontology class to generate a single sentence to introduce an entity. The two main realization processes, gender realization and active person realization, are described in Section 3.6.6.3 and 3.6.6.4 respectively. Section 3.6.6.5 describes the process of carrying out further realization for the relational patterns.

3.6.6.1 Search and Apply Patterns for Triples

The pattern search process looks for the best matching pattern given a triple with required metadata. The pattern search process prioritizes the modules in the order of occupational metonym patterns, verb frame patterns, relational patterns, and property patterns. Therefore, if a matching pattern is found at some stage, the framework will not execute the remainder of the pattern processing modules. The occupational metonymy patterns and verb frame patterns are prioritized as they are supervised lexicons which contain the validated lexicalization patterns. Then relational patterns are placed before property patterns because property patterns are generic templates and apply to many triples. However, such large scale application of same pattern can significantly decrease the language variety which is an essential feature of the language generation.

To apply a particular pattern processing module, the triple must satisfy the requirements specified by the pattern processing module. In essence, to apply the occupational metonym pattern, the predicate of the triple must be an occupational metonym or should be a generalization of a occupational metonym. In addition, the ontology class of the subject of the triple should match with the ontology class associated with the occupa-

tional metonym. This constraint is placed for the same reason described in Section 3.6.4 where the ontology class of the subject affects the lexicalization pattern, although triples share the same predicate. However, in a scenario where there is no exact ontology class hierarchy available, the framework will look for ontology class hierarchies from which the currently searching hierarchy is inherited. For instance, consider that the framework searches for a pattern for the *Agent* \rightarrow *Person* \rightarrow *Artist*, but there is no matching pattern for this ontology class. However, if there is a pattern with an ontology class hierarchy *Agent* \rightarrow *Person* and all other metadata is matched, then the framework will select this pattern to lexicalize the triple. This process is called RDF inference (Allemang and Hendler, 2008) where we look for the higher level ontology class hierarchy in case no match is found. The RDF inference is also used during searching relational patterns and property patterns which are discussed in the following paragraphs.

To apply CFG patterns the predicate should satisfy the constraints mentioned in Section 3.6.3 which include the predicate having the CFG form defined by the module.

When applying relational patterns to new triples we focus on multiple factors. Firstly, we look for a pattern which has the same ontology class hierarchy as the triple. We apply the RDF inference (Allemang and Hendler, 2008) in a scenario where no pattern is available for that particular ontology class hierarchy. Once a matching ontology class is found, we match the grammatical gender. This will only be accomplished for predicates which are already identified as gender specific (e.g., spouse, child, parent). The framework then matches the object multiplicity (explained in Section 3.6.4), if the predicate of the new triple has multiple objects. After matching the essential properties, the framework selects the lexicalization pattern that is available with the highest alignment score. However, if two patterns are available with all the required properties and with the same alignment score, then the relation confidence

Table 3.23 Examples of OWL CNL pattern based lexicalizations

Pattern base template	Example Lexicalizations
$\langle S?, is, O? \rangle_L$	$\langle Berlin, is, an administrative region \rangle_{LT}$ $\langle Vrije Universiteit, is, an educational institution \rangle_{LT}$ $\langle K2, is, a mountain \rangle_{LT}$ $\langle Rubens Barrichello, is, a formula one racer \rangle_{LT}$ $\langle Battlestar Galactica, is, a television show \rangle_{LT}$ $\langle Michael Jordan, is, a basketball player \rangle_{LT}$

score is consulted. Similarly, the framework compares the occurrence count, if all the aforementioned properties are similar.

The property patterns are applied if the predicate of the new triple is associated with a predetermined property pattern template. In addition, we also match the ontology class hierarchy. This module also supports the RDF inference as explained in the previous paragraphs.

3.6.6.2 Applying an Ontology based Controlled Natural Language Pattern

Each triple extracted from DBpedia contains the ontology class hierarchy that it belongs to. The core class of this hierarchy is the type that the triple subject belongs to. When developing a description it is important to describe the type of the triple collection. For example, if a description is generated for “Google”, then as the first sentence introducing “Google” is a company causes reader to follow the rest of the information easily. Furthermore, this supports the world knowledge based referring expression generation which will be discussed in Section 3.8. Table 3.23 depicts examples of ontology class based natural language patterns that will be added on top of the pattern search results.

Technically it is possible to associate any entity with a OWL CNL pattern as every entity must contain an ontology class hierarchy as metadata. However, in some cases describing an entity with a too broad class name can dramatically reduce the quality of

the generated text. Therefore, entities having a core ontology class as Agent, Person, and Place are not considered for OWL CNL patterns.

3.6.6.3 Gender Realization

Since relational patterns are extracted from the unstructured text and the pattern constraints are relaxed in case of a best matching pattern, there is a possibility that certain patterns can be selected with gender mismatches.

For example, consider the following scenario to understand the need of gender realization. Assume that triple $\langle \text{Lyndon B. Johnson, spouse, Lady Bird Johnson} \rangle_T$ which is categorized under the *OfficeHolder* ontology class is provided to be associated with a pattern. However, the relational pattern database contains only the lexicalization pattern $\langle S?'s, husband is, O? \rangle_L$ which can be associated with a subject having female grammatical gender (the grammatical gender of the subject entity can be retrieved from the grammatical gender database mention in Section 3.2.3.1). However, as the pattern search module drops the constraints in the absence of the best matching pattern relying on the latter realization steps, the above lexicalization pattern may be selected to lexicalize the above triple.

In scenarios such as above, we perform dependency parsing based realization steps to modify the lexicalization pattern to suit with the current triple. The pattern itself cannot be dependency parsed because it is a construct of three components and with expressions to denote subject and object. We first aggregate the three components of the pattern and transform it to a sentence like string. Next, $S?$ and $O?$ expressions are substituted with two proper nouns XSX and XOX . This is to avoid parser malfunction due to the question marks in the expressions.

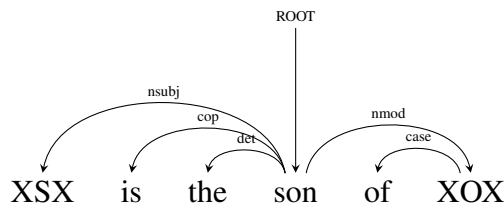


Fig. 3.15 Dependency parse result of the type-1 gender specific pattern

The core of this realization step is to identify the *nominal subject* dependency relation between the gender specific token and the object or subject. This dependency relation is analysed in two types which are explained in the following two sections.

The first type considers the gender token as the governing node. For example, consider the pattern $\langle XSX, is\ the\ son\ of,\ XOX \rangle_L$. This pattern cannot be applied to a triple like $\langle Elizabeth\ II, parent, George\ VI \rangle_T$ without any realization if there is not a best matching pattern for the triple which satisfies the grammatical gender constraint. Figure 3.15 shows the dependency parsed output of the aforementioned pattern. Based on this figure it is clear that the gender specific token is associated with the triple subject and forms gender mismatch if we apply this pattern to the given triple. To resolve this mismatch, we extract the gender token and query the masculine-feminine token database (introduced in Section 3.3.2) to retrieve the matching gender token. Upon receiving the result, the existing token is replaced with the correct gender token. This realization process is applied even if the triple object is present as the subject of the lexicalization pattern. For example, a pattern like $\langle XOX, is\ the\ son\ of,\ XSX \rangle_L$ is subjected to the same realization process. However, in such scenario the accuracy of the realization is based on whether the object value is present in the grammatical gender database. Since RDF does not enforce the object to appear as an entity (Hitzler et al., 2009), finding a record in a grammatical gender database related to a triple subject is not always guaranteed.

The second type considers the gender token as the dependent node. An example pattern $\langle XSX's\ son, is, XOX \rangle_L$ with a typed dependency is presented in Fig. 3.16.

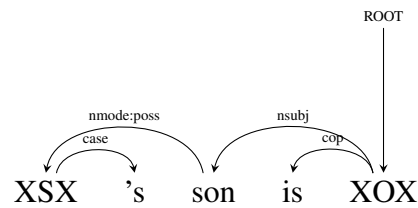


Fig. 3.16 Dependency parse result of the type-2 gender specific pattern

The main difference from the previous type is that now the gender token acts as the dependent of the relation. This case is also resolved similarly to the previous except for consideration of the relation direction. Furthermore, as in the previous type, subject and object interchangeability is also handled.

3.6.6.4 Active Person Realization

The pattern generation process produces patterns based on lexicons or through using unstructured text. None of these approaches place a special consideration on patterns related to people to check whether the specified person is alive, because if the person is not alive then the pattern should be changed accordingly. For instance, a pattern like $\langle S?, \textit{manages}, O? \rangle_L$ which is retrieved for the triple $\langle \textit{Steve Jobs}, \textit{director}, \textit{Apple Inc} \rangle_T$. Although, the pattern is semantically and syntactically correct, it is pragmatically incorrect as Steve Jobs is not alive. However, one option would be to use another constraint in the pattern generation level so that patterns related to ontology class hierarchies with "Person" are associated with a property to denote whether the pattern is derived from an active person or not. To successfully implement such a paradigm as in Section 3.6.6.3, there is a need for knowledge on predicates on which this constraint should be executed. However, in the current scenario it is difficult to derive such a predicate list due to the limitations from the Linked Data side. The limitation is that for some entities like business executives, their previous positions in companies are also shown with current ones and there is not any temporal data to detect whether that

person is not affiliated to the mentioned company. Due to absence of such metadata, it is not possible to introduce the constraint during the pattern generation phase.

3.6.6.4.1 Realization in OWL CNL Patterns

A OWL CNL pattern expresses the core class of the entity that is being introduced. If the entity belongs to the “*Person*” ontology class then the OWL CNL pattern must be realized based on whether the person is alive or not. This conversion is straightforward and converts the pattern directly to the past tense as in the below example through conversion of the copular verb.

$$\langle \textit{Barack Obama, is, Office Holder} \rangle_{LT} \implies \langle \textit{Barack Obama, was, Office Holder} \rangle_{LT}$$

3.6.6.4.2 Realization in Occupational Metonym Patterns

Since occupational metonyms are predefined pattern templates, they are realized by converting the verbal phrase to past tense. The example below shows the realization process of the lexicalization for the triple $\langle \textit{George VI, predecessor, Edward VIII} \rangle_T$.

$$\langle \textit{George VI, is preceded by, Edward VIII} \rangle_{LT} \implies \langle \textit{George VI, was preceded by, Edward VIII} \rangle_{LT}$$

3.6.6.4.3 Realization in Context Free Grammar Patterns

The context free grammar patterns are based on \mathcal{G} and based on the verb included in the predicate. These can be in two forms as mentioned in Section 3.6.3. The active voice past tense based patterns need not to be realized as they are already expressed in past tense. However, if passive voice form can be either present or past and it only depends whether person is alive or not. Therefore, the passive voice CFG patterns are realized to past tense accordingly.

3.6.6.4.4 Realization in Relational Patterns

Since the relational patterns are extracted from unstructured text through a fully au-

omatic process, we have no prior knowledge about their syntactic formation as in other modules which are based on predefined syntactic patterns. To address this lack of knowledge, we parse each pattern using dependency parser to identify the syntactic form of the pattern. A set of rules are then executed on each pattern to convert them to suit with inactivity. Table 3.24 shows the patterns that can be derived for realization and also mentions where realization is not necessary to carry out.

Table 3.24 Relational pattern realization for active person realization

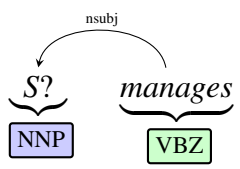
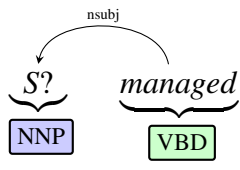
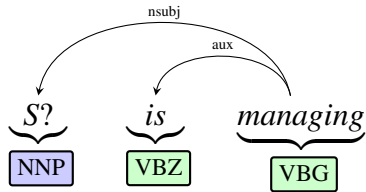
Pattern ID	Pattern with example	Realization needed?	Realization Result
LP-1	All patterns with <i>S?</i> 's in present tense (e.g., <i>S?</i> 's children are)	Yes	Direct replace
LP-2	All patterns with <i>S?</i> 's in past tense (e.g., <i>S?</i> 's children were)	No	–
LP-3		Yes	<i>S?</i> managed
LP-4		No	–
LP-5		Yes	<i>S?</i> was managing

Table continued on next page

Table 3.24 Relational pattern realization for active person realization (continued)

Pattern ID	Pattern with example	Realization needed?	Realization Result
LP-6	<p>S? (NNP) was (VBD) managing (VBZ)</p>	No	–
LP-7	<p>S? (NNP) is (VBZ) managed (VBZ)</p>	Yes	S? was managed
LP-8	<p>S? (NNP) was (VBD) managed (VBN)</p>	No	–
LP-9	<p>O? (NNP) is (VBZ) managed (VBN) by (IN) S? (NN)</p>	Yes	O? was managed by S?
LP-10	<p>O? (NNP) was (VBD) managed (VBN) by (IN) S? (NN)</p>	No	–

According to Table 3.24, out of the 8 different dependency patterns, only 5 require realization. The pattern LP-1 denotes a scenario where the subject has a possession

relationship with some object. Since the subject is not active currently, we convert this to past tense with a rule based approach converting the copula.

Out of the patterns, LP-3, LP-5, and LP-7 expresses the active present form of incident. In such a scenario, we extract the auxiliary verb and the root verb which is related to the subject and transform them to past tense. The pattern LP-9 deviates slightly from the rest and it is a passive form where the triple subject acts as the object of the pattern. In such a scenario, we convert the passive auxiliary verb to past tense.

3.6.6.4.5 Realization in Property Patterns

We also convert the property pattern types PP-1, PP-2, PP-3, and PP-4 to past tense if the person mentioned in the pattern subject is not alive.

3.6.6.5 Relational Pattern further Realization

The OpenIE based relation extractor can generate new relation tuples based on its trained examples. For example, extracting relation from the sentence “Michelle Obama, the wife of Barack Obama” will result in a relation tuple as $\langle \text{Michehlla Obama, be wife of, Barack Obama} \rangle_R$. Although the relation tuple is grammatically correct, it is not natural as generated by humans. To address this, our framework further realize these tuples targeting a version like $\langle \text{Michehlla Obama, is the wife of, Barack Obama} \rangle_R$. This is carried out by directly converting the relation segments of the relation tuples which has the aforementioned “*be*” relation pattern.

3.6.7 Output of the Lexicalization

The output of the lexicalization is a data structure containing all the above information which latter modules need to transform them to natural language paragraphs. Figure 3.17 illustrates the attribute-value matrix of a output produced by the lexicalization module.

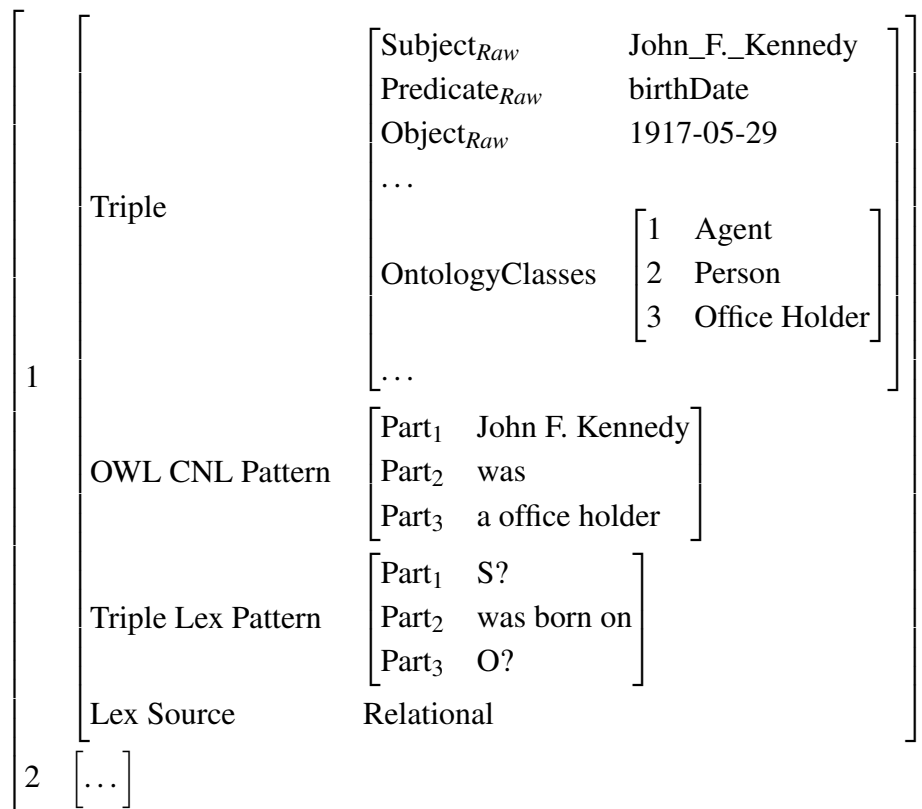


Fig. 3.17 Output of the lexicalization module depicted in an attribute-value matrix

3.7 Aggregation

The result of the lexicalization module is the natural language representation that can transform a triple to a natural language sentence. These sentences should then be aggregated to present a coherent natural language paragraph to the user. In this section we explain the architecture and the process of the aggregation module.

3.7.1 Overview of the Aggregation Process

Figure 3.18 depicts the overview of the architecture. We introduce the cluster aggregation model to generate natural language paragraphs from triples. In essence, the aggregation module executes a series of clustering steps based on pre-defined rules and then aggregates the sub-clusters to form natural language paragraphs from each

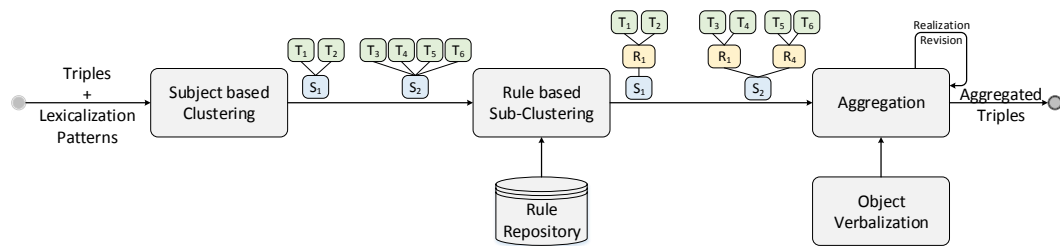


Fig. 3.18 Overview of the aggregation architecture

sub-cluster. However, the final result of this process does not represent the final natural language paragraphs, instead it is a hierarchical aggregation of triples which denotes the subject of the paragraph with the expression $S?$. This expression is resolved during the Referring Expression Generation (REG) which will be discussed in Section 3.8.

3.7.2 Subject based Clustering

The triple collection that aggregation module receives has no particular ordering or sorting. In this phase we first sort the triples, so that triples are clustered based on the subject. This clustering offers two advantages. Firstly, it makes each natural language passage centred towards a common theme which ultimately makes it easy for users to digest the information provided. Secondly, such clustering supports the REG to adhere with the Centering theory (Poesio et al., 2004) as all information related to the subject of the passage is provided as a single unit. Technically, the process scans the whole triple collection and clusters them into subjects. We then execute a triple ordering process to order them in the natural order that users prefer to read. To accomplish this we consult the ontology class - predicate database introduced in the Section 3.2.3.3 which contains the information on the order of predicates under a particular ontology class.

Table 3.25 shows an example of the clustering step for a sample question. This question and answer pair contains two entities, and triples which belong to these entities are clustered and ordered based on the predefined predicate priorities.

Table 3.25 A sample clustering and ordering of triples for the question “Who was the successor of John F. Kennedy?”

John F. Kennedy	Lyndon B. Johnson
$\langle \text{Lyndon B. Johnson, birth name, Lyndon Baines Johnson} \rangle_T$	$\langle \text{John F. Kennedy, profession, Politician} \rangle_T$
$\langle \text{Lyndon B. Johnson, birth date, 1908-08-26} \rangle_T$	$\langle \text{John F. Kennedy, birth date, 1917-05-28} \rangle_T$
$\langle \text{Lyndon B. Johnson, birth place, Texas} \rangle_T$	$\langle \text{John F. Kennedy, alma mater, Harvard College} \rangle_T$
$\langle \text{Lyndon B. Johnson, alma mater, Texas State University} \rangle_T$	$\langle \text{John F. Kennedy, spouse, Jacqueline Kennedy Onassis} \rangle_T$
$\langle \text{Lyndon B. Johnson, spouse, Lady Bird Johnson} \rangle_T$	$\langle \text{John F. Kennedy, party, Democratic Party} \rangle_T$

3.7.3 Rule based Sub-Clustering and Aggregation

Once subject based clustering is completed, we then carry out another clustering within each cluster. These sub-clusters correspond to an aggregated sentence, while former subject based cluster corresponds to a paragraph. The sub-clustering first assigns triples into their own clusters (forming clusters which exactly have only one triple) and then merge these clusters based on a set of rules. The rules are implemented considering both triples and the lexicalization patterns. The sections below explains the rules in detail.

3.7.3.1 Aggregation Rule - 1

This rule focuses on aggregating triples which have the same subject and predicate, but with different objects. In addition, the lexicalization patterns assigned for all the triples must be similar in content and the first part of the lexicalization pattern should hold the subject expression. For instance consider the below example.

Triple-1: $\langle \text{Nile River, source, Rwanda} \rangle_T$

Lex Pattern-1: $\langle S?, \text{flows through}, O? \rangle_L$

Triple-2: $\langle \text{Nile River, source, Egypt} \rangle_T$

Lex Pattern-2: $\langle S?, \text{flows through}, O? \rangle_L$

These triples are aggregated by aggregating the object components and keeping the rest of the lexicalization pattern components untouched. In essence the aggregated version of the above triples will result in the below tuple.

Aggregated sentence: $\langle S? \text{flows through Rwanda, and Egypt} \rangle_{AG}$

However, this aggregation can contain multiple triples and if all objects are displayed in a sentence, a user may lose interest due to long lists. Therefore, if there are more objects than a predetermined threshold we shorten the list (e.g, Rwanda, Egypt, and Ghana, among others). The threshold is currently set to three triples.

Another important factor is that the lexicalization pattern itself should support the object multiplicity to comply with this aggregation. Although, during the pattern search this is considered for relational patterns, property patterns are not associated with a metadata to identify whether they support multiplicity or not. Therefore, if triples are associated with property patterns, during this aggregation such patterns are transformed to their plural version.

3.7.3.2 Aggregation Rule - 2

There can be triples which convey information about multiple relations between two entities. Such triples contain the same subject and object, however, with different predicates. For instance, consider the below two triples.

Triple-1: $\langle \text{Harold and Maude, writer, Colin Higgins} \rangle_T$

Lexicalization Pattern-1: $\langle S?, \text{was written by}, O? \rangle_L$

Triple-2: $\langle \text{Harold and Maude, producer, Colin Higgins} \rangle_T$

Lexicalization Pattern-2: $\langle S?, \textit{was produced by}, O? \rangle_L$

The above lexicalized triples can be aggregated to reduce information repetition. We first analyse whether lexicalization patterns assigned to triples have the same structure which permit them to be aggregated. In essence the two patterns should contain the same number of tokens, and except one token all others should be stopwords and should be same in content. The token that is skipped must be associated with one of the predetermined POS tags: NN, VBN, VBD, and VBZ. The aggregation of aforementioned triples results in the following lexicalized triple.

Aggregated sentence: $\langle S? \textit{was written and produced by Colin Higgins} \rangle_{AG}$

The above examples focused on aggregating the triples associated with lexicalization patterns which has the subject expression as the first component. However, this aggregation rule is used to aggregate lexicalized triples which have the object expression as the first component as well. Since similarity in structure is compulsory, this can be accomplished in the same way as shown in the above example.

3.7.3.3 Aggregation Rule - 3

This aggregation focuses on shared language patterns in lexicalization patterns where all components are the same except for the ending preposition. For instance consider the below two scenarios.

Triple-1: $\langle \textit{Steve Jobs}, \textit{birth date}, \textit{1955-10-06} \rangle_T$

Lexicalization Pattern-1: $\langle S?, \textit{was born on}, O? \rangle_L$

Triple-2: $\langle \textit{Steve Jobs}, \textit{birth place}, \textit{New York} \rangle_T$

Lexicalization Pattern-2: $\langle S?, \textit{was born in}, O? \rangle_L$

These triples are aggregated under the common subject with a grammatical conjunction as below.

Aggregated sentence: $\langle S? \textit{was born on 1955-10-06 in New York} \rangle_{AG}$

3.7.3.4 Aggregation Rule - 4

This aggregation rule is a modification of the rule described in Section 3.7.3.1. The rule described in Section 3.7.3.1 requires subject expression to appear in the first part of the lexicalization pattern. Although this is appropriate for active voice based patterns, the subject expression should appear as the last item in a passive voice based pattern. The below scenario provides an example of this type of aggregation.

Triple-1: $\langle \textit{The Hound of the Baskervilles, author, Arthur Conan Doyle} \rangle_T$

Lexicalization Pattern-1: $\langle O?, \textit{was written by}, S? \rangle_L$

Triple-2: $\langle \textit{The Lost World, author, Arthur Conan Doyle} \rangle_T$

Lexicalization Pattern-2: $\langle O?, \textit{was written by}, S? \rangle_L$

Aggregated sentence: $\langle \textit{The Hound of the Baskervilles and The Lost World were written by } S? \rangle_{AG}$

The auxiliary verbs which appear at the start of the middle part of the pattern are also realized to plural tense to suit with the object aggregation. Further realization to comply with the aggregation is discussed in Section 3.7.4.

3.7.3.5 Aggregation Rule - 5

This rule aggregates the triples with properties which can be aggregated. Since it focuses only on properties, the lexicalization pattern associated with the triple must be a property pattern. The predicates that can be aggregated are provided as a database and a sample set of records is shown in Table 3.26. The below scenario shows an example of this type of aggregation.

Triple-1: $\langle \textit{Berlin, areaCode, 030} \rangle_T$

Lexicalization Pattern-1: $\langle S?'s \textit{area code, is, } O? \rangle_L$

Triple-2: $\langle \textit{Berlin, postalCode, 10001-14199} \rangle_T$

Lexicalization Pattern-2: $\langle S?'s \textit{postal code, is, } O? \rangle_L$

Table 3.26 Sample set of predicates that can be aggregated

Predicate groups	
areaCode, postalCode	weight, height
prominence, elevation	areaTotal, areaUrban, areaMetro
firstWin, lastWin	areaTotal, areaLand, areaWater
firstRace, lastRace	populationTotal, populationUrban, populationMetro
numberOfEpisodes, numberOfSeasons	populationDensity, populationUrbanDensity, populationMetroDensity

Aggregated sentence: $\langle S?'s \text{ area code, and postal code are respectively } 030, \text{ and } 10001-14199 \rangle_{AG}$

Although the example scenario is based on two triples, the approach can be extended to aggregate multiple triple into a single sentence.

3.7.3.6 Aggregation Rule - 6

This rule is the counterpart of the Rule-2 which aggregated the triples if subjects and objects of the triples are the same and the subject expression appears in the first part of the lexicalization pattern. The scenario below explains the aggregation through an example. Since both rules are focused on similar workflow, the actual implementation shares the same logic which modifies the middle part of the pattern.

Triple-1: $\langle \textit{The Gold Rush}, \textit{writer}, \textit{Charles Chaplin} \rangle_T$

Lexicalization Pattern-1: $\langle O?, \textit{was written by}, S? \rangle_L$

Triple-2: $\langle \textit{The Gold Rush}, \textit{director}, \textit{Charles Chaplin} \rangle_T$

Lexicalization Pattern-2: $\langle O?, \textit{was directed by}, S? \rangle_L$

Aggregated sentence: $\langle \textit{The Gold Rush was written and directed by } S? \rangle_{AG}$

3.7.3.7 Aggregation Rule - 7

This focused on aggregating lexicalizations where pluralizable tokens are present and share the same structure with subject expression as the first part in the pattern. For instance consider the following example.

Triple-1: $\langle \textit{Michael Jackson}, \textit{parent}, \textit{Joe Jackson} \rangle_T$

Lexicalization Pattern-1: $\langle S?'s, \textit{father is}, O? \rangle_L$

Triple-2: $\langle \textit{Michael Jackson}, \textit{parent}, \textit{Katherine Jackson} \rangle_T$

Lexicalization Pattern-2: $\langle S?'s, \textit{mother is}, O? \rangle_L$

Aggregated sentence: $\langle S?'s \textit{parents are Joe Jackson and Katherine Jackson} \rangle_{AG}$

In the aforementioned example we consult the pluralizable token records to identify that tokens *father* and *mother* can be pluralized by using the token *parent*. We use the dependency parsing based approach presented in Section 3.6.6.3 to identify the gender specific token which needs to be pluralized. The pluralization will be managed by the aggregation realization operations which are described in Section 3.7.4.

3.7.3.8 Aggregation Rule - 8

This rule is the counterpart of the Rule-7 which concentrates on patterns which have the object expression as the first part. An example scenario is shown below.

Triple-1: $\langle \textit{Steve Jobs}, \textit{child}, \textit{Lisa Brennan Jobs} \rangle_T$

Lexicalization Pattern-1: $\langle O?, \textit{is the daughter of}, S? \rangle_L$

Triple-2: $\langle \textit{Steve Jobs}, \textit{child}, \textit{Reed Jobs} \rangle_T$

Lexicalization Pattern-2: $\langle O?, \textit{is the son of}, S? \rangle_L$

Aggregated sentence: $\langle \textit{Lisa Brennan Jobs and Reed Jobs are the children of Steve Jobs} \rangle_{AG}$

3.7.4 Further Realizing the Aggregation

The further realization of the aggregation mainly focuses on resolving the pluralization mismatches in the triples aggregated through Rule-1 (see Section 3.7.3.1) or Rule-4 (see Section 3.7.3.4) For instance, consider the below two triples and the associated lexicalization patterns.

Triple-1: $\langle \textit{Lyndon B. Johnson, child, Luci Baines Johnson} \rangle_T$

Lexicalization Pattern-1: $\langle S?'s, daughter\ was, O? \rangle_L$

Triple-2: $\langle \textit{Lyndon B. Johnson, child, Lynda Bird Johnson Robb} \rangle_T$

Lexicalization Pattern-2: $\langle S?'s, daughter\ was, O? \rangle_L$

The above triples can be aggregated using the Rule-1. However, the aggregated lexicalization patterns need to be corrected as they only focus on a single object. Since the pattern depicts the possession of an object, these patterns can be directly pluralized by pluralizing the noun phrase (e.g., *daughter*) which shows the possession. This will result in an aggregation as $\langle S's\ daughters\ were\ Luci\ Baines\ Johnson\ and\ Lynda\ Bird\ Johnson\ Robb \rangle_{AG}$. However, such direct transformations are very rare and in many cases we need to identify the exact noun which needs to be pluralized. For this we employ the dependency parsed lexicalization pattern to identify the syntactic relations among the tokens. Table 3.27 shows the dependency rules that we use to determine the token to be pluralized. Furthermore, the framework does not pluralize the tokens which are not pluralizable. Such tokens are identified using the POS tagging and tokens which have the POS tags such as *VBN*, *VBP*, and *VP* are not pluralized.

We use the method proposed by Conway (1998) to pluralize the candidate token identified through the dependency rules. Although a simple pluralization algorithm can be implemented as depicted in Listing 3.7, such an algorithm will fail due to the idiosyncratic nature of English. The pluralization approach presented by Conway (1998) incorporates three types of pluralizations strategies: universal defaults, general suffix-

Table 3.27 Dependency rules to identify the token to be pluralized

	Rule-1	Rule-4
Dependency pattern example		

based rules, and exceptional cases. The universal rules cover the well-known principles of pluralization such as appending *-s* to nouns. However, the universal rules are applied only when other rules are inapplicable. The suffix categories focus on pluralization based on a particular word suffix. For example, nouns ending in *-ss* becomes *-sses* when transforming to plural form. In addition to these forms, there are other exceptional scenarios. For example, when transforming words *trilby* and *ox* to plural form, they become *trilbys* and *oxen* respectively.

```

1 def plural (word)
2   if word.endswith('y'):
3     return word[:-1]+'ies'
4   elif word[-1] in 'sx' or word[-2:] in ['sh', 'ch']:
5     return word+'es'
6   elif word.endswith('an'):
7     return word[:-2] + 'en'
8   else:
9     return word+'s'

```

Listing 3.7 A naive approach to pluralize English words

Although, the above pluralization can successfully pluralize tokens individually, when applying to the aggregation it requires additional exceptional scenarios. For instance, consider the below example where triples are applied with lexicalization patterns which need to be aggregated to form a single sentence.

Triple-1: $\langle \textit{Margaret Thatcher}, \textit{child}, \textit{Carol Thatcher} \rangle_T$

Lexicalization Pattern-1: $\langle S?'s, \textit{daughter was}, O? \rangle_L$

Triple-2: $\langle \textit{Margaret Thatcher}, \textit{child}, \textit{Mark Thatcher} \rangle_T$

Lexicalization Pattern-2: $\langle S?'s, \textit{son was}, O? \rangle_L$

The pluralization algorithm presented by Conway (1998) pluralizes the tokens considering their individual usage. However, to suit with the pluralization required for aggregations mentioned in Section 3.7.3.7 and Section 3.7.3.8, certain exceptions are needed. For example, when pluralized *daughter* and *son* should become *children*. The framework handles these pluralizations using a set of exceptions which cover special scenarios.

3.7.5 Output from the Aggregation

Figure 3.19 depicts an example aggregation output in the proto-phrase specification. The output is a collection of aggregation clusters that each carry information related to the entities mentioned in the question/answer and previous outputs (e.g., lexicalizations) related to these entities.

3.8 Referring Expression Generation

The aggregation phase discussed in Section 3.7 keeps the subjects expressions untouched. The objective of this is to assign referring expressions when generating a paragraph of text without repeating the verbalized subject. A referring expression is a noun phrase that can identify an entity (i.e. subject of the paragraph) which is already mentioned. We focused on two factors when generating referring expression for the informative answers. Firstly, we used variations of referring expression rather being bound to one particular type. This is to increase the language variety which is an

1	Subject _{Raw}	Steve_Jobs	
	Subject _{Verbalized}	Steve Jobs	
	OntologyClasses	$\begin{bmatrix} 1 & \text{Agent} \\ 2 & \text{Person} \end{bmatrix}$	
	NaturalGender	Male	
	SubClusters	1	$\begin{bmatrix} \text{LexicalizationOutputs} & \begin{bmatrix} 1 & \dots \\ 2 & \dots \\ \dots & \dots \end{bmatrix} \\ \text{AggregationRule} & \text{RULE_3} \end{bmatrix}$
		2	$\begin{bmatrix} \dots \end{bmatrix}$
	AggregatedSentences	1	S? was born on February 24, 1955 in California
		2	S? was the founder of Apple Inc.
		...	
	2	$\begin{bmatrix} \dots \end{bmatrix}$	
...			

Fig. 3.19 A sample output from the aggregation module

essential factor of human produced language. It is also important to make sure that ambiguity does not exist between multiple entities for which the framework generated descriptions. This is already addressed by the design of the aggregation framework which supported the Centring theory as described in Section 3.7.2.

We first classify the entities into humans and things, as referring expressions used in these two categories vary significantly. This classification is carried out using the ontology class hierarchies. In essence, all the entities that belong to the “*Person*” ontology class are classified as humans and the rest are classified as things.

The entities classified as “*Person*” are assigned two types of referring expressions: pronominal anaphora (e.g., he, she, him, her, hers, his) and first name of the person. In the first sentence we use the actual entity name as it appears in the DBpedia triple. From the next sentence onwards we use the pronouns and first name interchangeably. In essence if the same referring expression is expected to appear in more than two

consecutive occurrences, then in the third occurrence the referring expression is changed to a semantically similar different form. This setting is designed targeting a language variety rather using the same referring expression for the whole paragraph.

The entities that are classified as “*Things*” are assigned three types of referring expressions: pronominal anaphora (e.g, it, its), the name of the entity (e.g., Google, Google’s), and core ontology class (e.g., company, airline). As described earlier, these referring expressions are also used interchangeably to maintain the language variety. Furthermore, using the core ontology class opens the possibility of integrating the world knowledge based referring expression in the generated text. However, all core ontology classes are not suitable as referring expressions. Therefore, only a selected set of 10 ontology classes (e.g., river, book, company, film) are used as referring expressions.

3.9 Structure Realization

Structure realization is the final stage of the RealText framework and it is generally considered as the final step of any NLG based application. Structure realization concentrates on ordering the entity descriptions and presenting the generated text in different formats.

3.9.1 Entity Description Ordering

As the first task of structure realization, the framework reorders the entity description based on the communal common ground principle (Clark and Brennan, 1991). The communal common ground principle is a key principle in communication in which it states that conversations of groups working together are grounded on achieving common ground or mutual knowledge. Applying this principle in our presentation framework, we specify that entities mentioned in the question should be prioritized. This is mainly

because the person asking the question has some prior knowledge on entities that he/she mentions in the question, however, the answer may contain a completely new entity which is unfamiliar to him/her.

3.9.2 Presentation Formats

The RealText framework presents generated text in six different formats, namely, Speech Synthesis Markup Language (SSML), Hypertext Markup Language (HTML), L^AT_EX, Open Document Format (ODF), RDF, and Extensible Markup Language (XML). The following sections discuss the process of transforming text to these formats and the importance of these different presentation methods.

3.9.2.1 Speech Synthesis Markup Language (SSML)

The SSML is the document format that focuses on providing text to speech synthesis programs with annotations. The speech synthesizer uses these annotations to decide how each token, phrase, sentence or paragraph should be transformed into voice. To support QA systems which are enabled with speech utilities, we provide the SSML version of the generated answer. Since the framework builds the informative answer from the information units (i.e., triples), we get a higher freedom of annotating information appropriately than traditional answer presentation models which rely on summarization and sentence extraction. In addition, as we select information from the Semantic Web, the information is associated with metadata to support the annotation.

The annotating procedure is applied to both answer sentences as well as to the informative answers. In the answer sentence, we annotate the answer by emphasizing it. In informative answers, annotations are applied for date and ordinal numbers as identified through predicate requirements mentioned in Section 3.2.3.5, Section 3.2.3.6, and Section 3.2.3.2. An example of annotated answer is shown in Listing 3.8.

```

1 <paragraph>
2   <sentence>The Mount Everest is <emphasis><say-as interpret-as="
      cardinal">8848m</say-as></emphasis> high.</sentence>
3 </paragraph>
4
5 <paragraph>
6   <sentence>Mount Everest is a mountain. </sentence>
7   <sentence>It is the highest summit in Tibet Autonomous Region ,
      China , and Tingri County , among others . </sentence>
8   <sentence>Its prominence is <say-as interpret-as=" cardinal ">8848.0
      m</say-as>. Its elevation is <say-as interpret-as=" cardinal ">
      8848.0 m.</say-as> </sentence>
9   <sentence>It was climbed in <say-as interpret-as=" date ">1953</
      say-as>. </sentence>
10  <sentence>It was climbed by Edmund Hillary , and Tenzing Norgay.</
      sentence>
11 </paragraph>

```

Listing 3.8 SSML annotated answer for the question “How high is the Mount Everest?”

3.9.2.2 Extensible Markup Language (XML) and Hypertext Markup Language (HTML)

The generated answers are also presented in XML and HTML formats. The XML allows the framework to communicate with another system while HTML is used as a pure presentation mechanism. Listing 3.9 and Listing 3.10 show examples of presentation scenarios for both XML and HTML formats respectively.

3.9.2.3 \LaTeX

The answer is also presented in \LaTeX format by programmatically converting the answer to the \LaTeX source. Listing 3.11 shows an example of the formatted answer.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <answerSentence>The capital of Canada is Ottawa</answerSentence>
3 <entityDescription label="Canada"> Canada is a country. Ottawa is
   the capital city of it. Largest city in it is Toronto. Canadian
   dollar is the official currency of Canada. Its anthems are God
   Save the Queen, and O Canada. Its population density is 3.41ppkm2
   . Canada's area total is 10 trillion m2. Beverley McLachlin is
   the Prime Minister of it. It has influenced Canadian music.
4 </entityDescription>
5 <entityDescription label="Ottawa"> Ottawa is a city. It is the
   capital of the Canada. It is part of National Capital Region ,
   and Ontario. Ottawa is founded on January 01, 2001. Jim Watson
   had been the last mayor to amalgamation of it. Its area total ,
   area urban , and area metro are respectively 2.78 billion m2 ,
   501.92 million m2 , and 5.716 billion m2. Its area code is 613.
   Ottawa's elevation is 70.0 m. Its population total , and
   population density are respectively 883391, and 316.6ppkm2. Its
   population urban density is 1860.1ppkm2. Ottawa's population
   metro is 1.24 million. Its population urban is 933596. Its postal
   code is K1A–K4C. Ottawa's time zone is Eastern Time Zone.
6 </entityDescription>
```

Listing 3.9 XML formatted answer for the question “What is the capital of Canada?”

3.9.2.4 Open Document Format (ODF)

Open Document Format (ODF) is the most widely used document format based on the XML specification. The answers are also formatted to the ODF format through the JOpenDocument library. The answer will be printed as formatted plain text as seen in a normal document.

3.9.2.5 Resource Description Format (RDF)

This framework generates answers by extracting information from the Linked Data which is essentially a collection of RDF triples. The objective of transforming the

```

1 <html>
2   <body>
3     <h2><b>Answer Sentence</b>: The capital of Canada is Ottawa</h2>
4     <b>Entity Descriptions:</b>
5     Canada is a country. Ottawa is the capital city of it. Largest
6     city in it is Toronto. Canadian dollar is the official currency
7     of Canada. Its anthems are God Save the Queen, and O Canada ...</
8     br>
9     Ottawa is a city. It is the capital of the Canada. It is part of
10    National Capital Region , and Ontario. Ottawa is founded on
11    January 01, 2001. Jim Watson had been the last mayor to
12    amalgamation of it ... .
13  </body>
14 </html>

```

Listing 3.10 HTML formatted answer for the question “What is the capital of Canada?”

```

1 \documentclass[10pt,a4paper]{article}
2 \usepackage[latin1]{inputenc}
3 \begin{document}
4 \textbf{Answer Sentence}: The capital of Canada is Ottawa
5 \textbf{Entity Descriptions}:
6 Canada is a country. Ottawa is the capital city of it. Largest city
7   in it is Toronto. Canadian dollar is the official currency of
8   Canada. Its anthems are God Save the Queen, and O Canada ... .
9
10 Ottawa is a city. It is the capital of the Canada. It is part of
11   National Capital Region, and Ontario. Ottawa is founded on
12   January 01, 2001. Jim Watson had been the last mayor to
13   amalgamation of it ... .
14 \end{document}

```

Listing 3.11 L^AT_EX formatted answer for the question “What is the capital of Canada?”

generated answers back to RDF is to enrich the Linked Data cloud by adding an enriched answer collection as RDF triples. This may stand as an initiative for similar

QA systems do not provide their outputs in simple text that only humans can understand, but a collection of triples that machines can understand and further process. Listing 3.12 shows an example of the generated answer in RDF format where predicates are given from a predetermined ontology. Furthermore, at this stage of research we have defined a limited ontology to organize the questions based on the entities which are mentioned in the question.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <rdf:RDF
3   ...
4   xmlns:rtprop="http://realtext.org/property/">
5   <rtprop:answerSentence xml:lang="en">
6     The capital of Canada is Ottawa
7   </rtprop:answerSentence>
8   <rtprop:entityLabel xml:lang="en">
9     Canada
10  </rtprop:entityLabel>
11  <rtprop:entityLabel xml:lang="en">
12    Ottawa
13  </rtprop:entityLabel>
14  <rtprop:entityDescription xml:lang="en">
15    Canada is a country. Ottawa is the capital city of it. Largest
16      city in it is Toronto. Canadian dollar is the official currency
17      of Canada. Its anthems are God Save the Queen, and O Canada ...
18  </rtprop:entityDescription>
19  <rtprop:entityDescription xml:lang="en">
20    Ottawa is a city. It is the capital of the Canada. It is part of
21      National Capital Region, and Ontario. Ottawa is founded on
22      January 01, 2001. Jim Watson had been the last mayor to
23      amalgamation of it ...
24  </rtprop:entityDescription>
25 </rdf:RDF>
```

Listing 3.12 RDF formatted answer for the question “What is the capital of Canada?”

3.10 Chapter Summary

This chapter detailed the methodology of our proposed framework for generating informative answers for QA systems. We first described the architecture of the framework which is comprised of an answer sentence generation component and an entity description generation component which together produces an informative answer. Both the components were explained in detail focusing on their sub components which construct the intended output. Specifically, the entity description sub components were described in terms of three main modules, namely, lexicalization, aggregation, and referring expression generation. The sentence was generated corresponding to the answer and the entity descriptions were then passed to the structure realization module which formats the answer in a number of different formats, so that it could be used as input to a variety of applications.

The next chapter explains the evaluation strategy that was employed to test the methodology proposed in this chapter.

Chapter 4

Evaluation

This chapter discusses the details of the evaluation of the RealText framework. The evaluation mainly focuses on human evaluation in which the evaluators rate the generated answer texts based on a criteria. In addition, an investigation was also carried out into the effectiveness of using automatic metrics for the task of natural language evaluation. We used four different metrics to explore whether any of them have any correlation with the human evaluation.

The rest of the chapter is structured as follows. Section 4.1 introduces the test dataset used for the evaluation. In Section 4.2, we provide detailed statistics focusing on individual modules of the framework. Section 4.3 discusses the evaluation settings and the results from the human evaluation and Section 4.4 discusses the automatic metric based evaluation which was carried out as an investigative study. Section 4.5 and Section 4.6 discuss comparisons with other works and identifies the shortfalls of the research. We conclude the chapter with a summary in Section 4.7.

4.1 The Test Dataset

For the evaluation phase of the framework, we used 52 factoid questions from QALD-2 test dataset which contains 100 questions. We eliminated the list based questions, invalid questions, and questions which are stated as imperative constructs which are not based on interrogative words (see Section 3.5.2 for further explanation). Table 4.1 provides the details on the statistics of the test dataset. The entities mentioned in the test dataset can be categorized into 37 ontology classes, however, the entities were not equally distributed among these ontology classes. The classes such as Office Holder, Country, and Administrative Region are the three classes with the highest number of entities. This was mainly because the entities related to these classes were mentioned in a large number of questions, although the main intent entity of the questions were different. For instance, in the question “Who is the governor of Texas^{[Administrative Region]?}” has the answer as the main intent, which is “Rick Perry^{[Office Holder]”.} However, it also contains an entity in the “Administrative Region” ontology class. Table 4.1 summarizes the overall statistics on triples in the test dataset. The table shows that from 3381 triples, 1984 were invalid hence were filtered out in the preprocessing phase. The invalid triples were those that were inappropriate for lexicalization such as URL links to external resources (e.g., Wikipedia links, image links, VIAF ID).

Figure 4.1 shows classification of the triples into 52 test questions. The graph shows over 30% of the questions had more than an average of 27 triples for each of the questions in the test dataset. The questions which deal with more informative entities (entities which have a significantly higher amount of information than others) such as *Lyndon B. Johnson*, *John F. Kennedy*, and *Margaret Thatcher* were having 129.62%, 66.67%, and 125.92% higher than the average triple count. However, the valid triple count does not necessarily determine the descriptiveness of the answer. This is because

Table 4.1 Statistics of the test dataset

Factor	Value
Number of unique ontology classes	37
Number of unique DBpedia entities	80
Number of triples	3456
Number of valid triples	1421
Number of invalid triples	2035
Number of DBpedia entities of type Masculine	30
Number of DBpedia entities of type Feminine	8
Number of DBpedia entities of type Neutral	42

the descriptiveness of the answers are based on the number of triples lexicalized by the lexicalization module.

From the total entity collection, 52.5% were identified as having neutral gender while 37.5% and 10% were identified as masculine and feminine entities respectively. The gender of the entities is used in later processes such as lexicalization and referring expression generation significantly to generate correct gendered text forms.

From the entire question set, only 37 questions had shared ontology classes. Ontology classes such as “Administrative Region” and “Country” are used in a number of questions as a part of the question. For example, the question “Who is the governor of Texas?” is expecting the answer “Rick Perry” which is associated with the “Office Holder” ontology class while “Texas” belongs to the “Administrative Region” ontology class. On the other hand, only four questions had shared entities where two entities were shared between the questions.

Table 4.2 offers insights into the statistics related to the valid triple collection. The table lists the statistics on the unique triples as some questions have overlapping entities. From the valid triple collection 56.50% of triples had predicates with multiplicity property. A triple is associated with multiplicity property if another triple or triples exist in the triple collection with the same subject and predicate, but with different

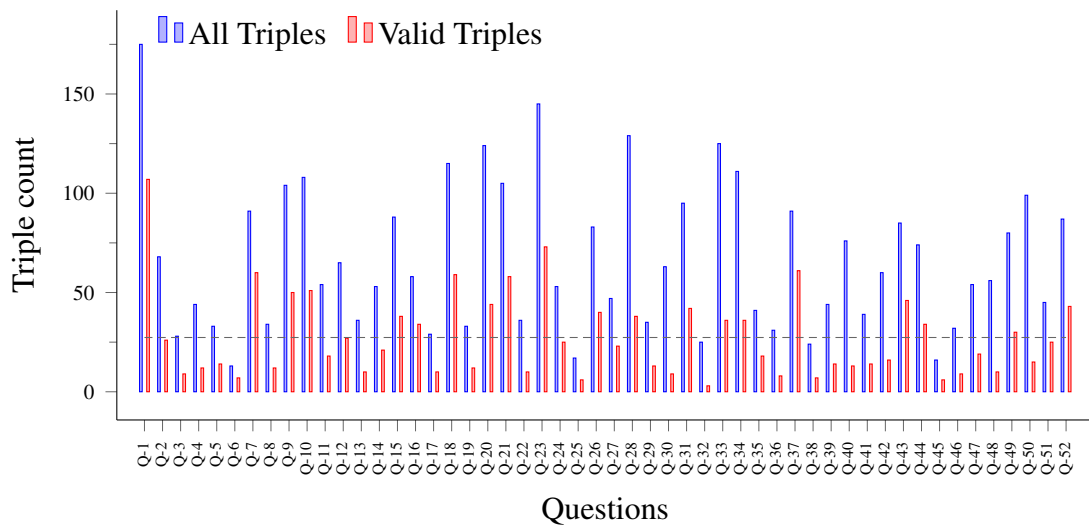


Fig. 4.1 Question wise classification of triples. The grey coloured dash line represents the average of the valid triples.

object values. For instance, children of Margaret Thatcher can be shown in two triples as $\langle \text{Margaret Thatcher}, \text{child}, \text{Mark Thatcher} \rangle_T$ and $\langle \text{Margaret Thatcher}, \text{child}, \text{Carol Thatcher} \rangle_T$, where both triples have the same subject and predicate, but with different object values (i.e., Mark Thatcher, Carol Thatcher). The triple collection also contained 17.31% of triples with predicates which needed date as an object. These included predicates such as *birth date* and *death date* which require single date as the object as well as the predicates such as *term period* which is a date range. Triples with measurement units as the object constituted only 8.65% from the valid triple collection. However, a question set analysis showed that 46.15% (24 questions) of the entire question set contained at least one measured number predicate which makes this an important category to lexicalize. Furthermore, the triple collection also had 3.43% triples which comprised of predicates which required normal numbers.

Table 4.2 Statistics of the Valid Triples

Factor	Value
Number of unique triples with multiplicity	790
Number of unique triples with date predicates	242
Number of unique triples with measured number predicates	121
Number of unique triples with normal number predicates	48

4.2 Module wise Evaluation and Statistics

This section describes the evaluation related to the individual modules of the RealText framework. It is divided into four sections corresponding to the four modules from the framework.

4.2.1 Answer Sentence Generation

Table 4.3 provides the statistics of the extracted patterns. A total of 25 unique patterns were extracted from the QALD-2 development dataset, out of which 72% were wh-interrogative patterns and 28% were polar interrogatives. This development dataset contained 41 wh-interrogatives and 8 polar interrogatives. As explained in Section 3.5.3, the answer sentence generation patterns are based on dependency paths originating from the root of the dependency parsed question, which is the subtree extracted from the complete dependency tree of the parsed question. The maximum number of dependency paths of any identified subtree was reported as four, where only five wh-interrogatives and one polar interrogative contained four dependency paths (originated from root) in the entire QALD-2 development dataset. An example pattern with four dependence paths is shown in Table 4.4. The table also shows the parsed source question used to extract the pattern. As shown in Table 4.4, the pattern contains various dependency relations originating from the root to various linguistic components, namely, auxiliary verb, adverb modifier, nominal subject, and direct object.

Table 4.3 Statistics on the answer sentence generation patterns

Factor	Value
Extracted unique wh patterns	18
Extracted unique polar patterns	7
Maximum Root oriented Dependency Paths (wh)	4
Minimum Root oriented Dependency Paths (wh)	2
Maximum Root oriented Dependency Paths (polar)	4
Minimum Root oriented Dependency Paths (polar)	2
Average of Unique Root oriented Dependency Paths (wh)	3
Average of Unique Root oriented Dependency Paths (polar)	3

Table 4.4 An example of four dependency path based answer sentence generation pattern

Original Typed Dependency Tree	Typed Dependency Subtree	Dependency Pattern
<p>Original Typed Dependency Tree for the sentence "When did Finland join the EU?". The root node is labeled "ROOT". Arrows indicate dependencies: ROOT to "When" (advmod), ROOT to "did" (aux), ROOT to "Finland" (nsubj), ROOT to "join" (doobj), ROOT to "the" (det), and ROOT to "EU?" (R).</p>	<p>Typed Dependency Subtree for the sentence "When did Finland join the EU?". The root node is labeled "ROOT". Arrows indicate dependencies: ROOT to "X[wh]" (advmod), ROOT to "X" (aux), ROOT to "X" (nsubj), ROOT to "R" (doobj), and ROOT to "X" (det).</p>	<p>nsubj ↔ aux+Root ↔ dobj ↔ advmod</p>

The minimum dependency relations originating from the root for wh- and polar interrogatives was reported as two. These dependency structures connected only two linguistic components such as a nominal subject with a direct object or copular with a nominal subject. The majority of the parses contained three dependency relations for both wh- and polar interrogatives.

It is also important to mention that the development dataset contained questions which had a variety of dependency paths. However, the proposed pattern extraction approach considers only the dependency subtree which is based on the paths originating from the root of the dependency tree. Therefore, the rest of the dependency paths (dependency paths that do not originate from root) were not considered for the pattern extraction task. For instance, the dependency parsed question in Fig. 4.2 has seven

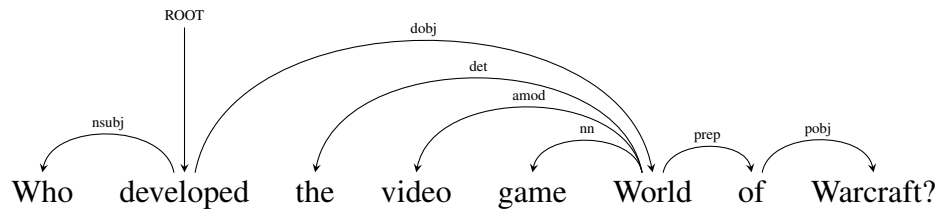


Fig. 4.2 Question in development set which contains seven dependency relations. The root oriented subtree can classify this question using two unique dependency paths for answer sentence generation.

Table 4.5 Answer sentence generation basic statistics

Factor	Value
Wh interrogatives processed	44
Polar interrogatives processed	8
Applied Wh patterns	11
Applied Polar patterns	1
Measurement unit embedded sentences	3
Revised periphrastic tenses	7

dependency relations between the eight tokens in the question. So for this example only two of the dependency paths originating from the root (*nsubj* and *dobj*) were used in the pattern extraction process.

Table 4.5 shows the statistics for the answer sentence generation task for the test dataset. Out of the 52 questions in the test dataset, 44 questions (84.61%) were wh-interrogatives and the rest (8 questions representing 15.38% of the dataset) were polar interrogatives. Out of the 25 extracted patterns, 11 wh- and one polar patterns were able to generate syntactically and semantically accurate answer sentences for 41 out of the 52 questions in the test set, giving us an accuracy of approximately 78.84%.

Some questions required measurement units embedded into the answer sentence as described in Section 3.5.5. For example, a question such as “*How tall is Michael Jordan?*” requires the measurement unit of the answer, which is “*meters*”, to be embedded when generating the answer sentence. The answer sentence generation

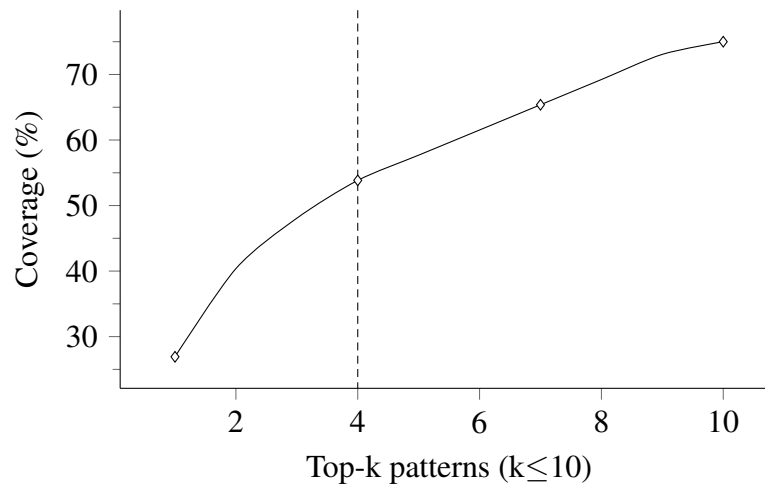


Fig. 4.3 Coverage of the extracted patterns in the test dataset

module embedded measurement units for three answer sentences generated for the test dataset. These answer sentences required properties such as height and temperature to be associated with required measurement units. The process also revised seven periphrastic tenses during the answer sentence generation. These revisions included the conversion of *does have* ⇒ *has*, *did marry* ⇒ *married*, *does come* ⇒ *comes*. All the past participle forms required for these verbs were present in the verb information database developed based on the VerbNet.

Figure 4.3 shows the coverage of the top-10 patterns in the test dataset. According to the graph, the first four patterns out of the 25 patterns, were able to correctly cover more than 50% of the test questions. This affirms that the patterns identified are highly representative and can be generalized to generate answer sentences in new question sets. Table 4.6 shows the top five patterns, sorted in descending order according to the number of questions covered. According to this listing, nominal subject and copular verb related dependency pattern achieved the highest representation by generating answer sentences for 14 questions in the test question set. The second pattern (nominal subject with direct object pattern) was able to generate answer sentences for seven test questions.

Table 4.6 Top-5 Patterns in answer sentence generation

ID	Top 5 Patterns
1	nsubj↔Root[wh]↔cop
2	nsubj[wh]↔Root↔doobj
3	nsubj↔Root+aux↔doobj[wh]
4	nsubjass[wh]↔Root↔auxpass↔prep
5	nsubj↔Root↔dep[wh]

Table 4.7 Active and passive form of the subtree patterns

	Active form	Passive form
Parsed question	<p>Where did Abraham Lincoln die?</p>	<p>When was Capcom founded?</p>
Subtree pattern	nsubj ↔ aux+Root ↔ advmod	nsubjpass ↔ auxpass+Root ↔ advmod

Additionally, some of the identified patterns also had their passive version in the pattern set. Table 4.7 shows an example scenario where active and passive forms of the subtree patterns with two example questions. The two patterns depicted in the Table 4.7 are the active form and the corresponding passive form, which were derived from the two dependency parsed questions in the table. Although such relationship existed between different patterns, the current answer sentence generation module did not include the conversions since the patterns were derived entirely from the training set. This is mainly because the pattern generation was based entirely on the training data set while the test set was used only for the purpose of evaluation in order to be able to arrive at a reliable performance figure.

The answer sentence generation module was not able to generate sentences for 11 questions in the test dataset. This was because the dependency parsed question contained patterns which were not found in the training dataset. Table 4.8 shows

Table 4.8 Example question for which an answer sentence was not generated due to the absence of patterns

Dependency parsed question	Pattern
	nsubj ↔ cop ↔ det ↔ Root ↔ prep[wh]

an example for which an answer sentence was not generated due to the absence of appropriate patterns.

4.2.2 Lexicalization

The lexicalization process consisted of four pattern processing modules, namely, occupational metonym patterns, CFG patterns, relational patterns, and the property patterns. In this section, we first focus on the statistics of the pattern processing modules and then move to the application of these patterns and their accuracies in the test question set.

4.2.2.1 Statistics of Pattern Processing Modules

4.2.2.1.1 Occupational Metonym Patterns

Occupational metonym patterns utilized a lexicon which is comprised of metonyms related to occupations (e.g., commander, teacher). The lexicon is comprised of 33 occupational metonyms which were associated with a pattern. Table 4.9 shows a sample set of occupational metonym lexicon for illustration. These metonyms were used to transform a triple into natural language when the triple contains an occupational metonym as the predicate. As can be seen from the last three records of the Table 4.9, the same occupational metonym (i.e., Author) can exist in two ontology class hierarchies.

Table 4.9 Selected set of records from the occupational metonym lexicon

Metonym	Ontology class hierarchy	Pattern		
		Part 1	Part 2	Part 3
Director	Work → Film	S?	was directed by	O?
Predecessor	Agent → Person	S?	is preceded by	O?
Publisher	Work → Written Work	S?	was published by	O?
Commander	Event → Societal Event → Military Conflict	S?	was commanded by	O?
Owner	Agent → Organization	S?	is owned by	O?
Author	Work → Website	S?	was created by	O?
Author	Work → Art Work	S?	was painted by	O?
Author	Work → Software	S?	was developed by	O?

However, as discussed in Section 3.6.2, the lexicalization patterns associated with occupational metonyms vary based on the ontology class hierarchy of the subject which was taken into account by our framework within the appropriate ontology class hierarchy when selecting a pattern triple.

4.2.2.1.2 Context Free Grammar Patterns

The Context Free Grammar (CFG) pattern module did not contain any pattern lexicon (such as in occupational metonym patterns and property patterns) or a pattern generation process (like relational patterns). This module works dynamically by associating the triples with the patterns by checking the requirement rule as described in Section 3.6.3. The detailed evaluation of how the individual CFG patterns contributed to the lexicalization is described in Section 4.2.2.2.

4.2.2.1.3 Relational Patterns

The relational patterns for the processing module extracts the patterns by aligning relations extracted from the unstructured text with the triples. This section describes the statistics for the test data by the pattern extraction process. Table 4.10 shows the statistics related to the pattern generation process of the relation pattern processing

Table 4.10 Statistics of the relational pattern generation process

Factor	Value
Number of sentences processed	17021
Number of extracted relations	56112
Number of adjectives processed	1091
Number of adverbs processed	536
Number of compound nouns processed	108

module. The module processed 17021 sentences contained in 144 text files which had been processed for co-reference resolution which resulted in an average of 118 sentences per text file. Using these sentences OpenIE module described in Section 3.6.4 was able to extract 56112 relations in the range between 3 and 4 relations per sentence. This shows that the OpenIE module has contributed to the relation extraction extensively compared to ClosedIE as described in its theoretical foundation Etzioni et al. (2008). This high level of relation extraction can be attributed to the relational phrase based approach that the OpenIE model uses compared to the traditional ClosedIE information extraction paradigm. During the process of extracting the relational pattern by aligning them with the triples, the model processed 1091 adjectives, 536 adverbs, and 108 compound nouns. This strategy of processing supports pattern extraction to derive a more cohesive pattern rather than simply aligning and extracting a pattern from the relation.

The statistics for the extracted relational patterns are shown in the Table 4.11. The OpenIE extracted 56112 relations, from which 1871 unique patterns were devised. The pattern extraction process resulted in 1871 unique patterns (from the relation collection) because the relations could only be extracted if the triple contained both the components (i.e., subject and object) in alignment with triples and therefore became candidates for pattern extraction. The OpenIE extracts all the relations that are present in a sentence. However, only some of them have both the triple subject and object included which

Table 4.11 Statistics of the extracted relational patterns

Factor	All	Above the threshold
Number of extracted unique relational patterns	1871	391
Number of unique relational patterns with feminine grammatical gender	144	76
Number of unique relational patterns with masculine grammatical gender	835	188
Number of unique relational patterns with neutral grammatical gender	885	127
Number of unique relational patterns with unknown grammatical gender	7	0
Number of unique multiplicity patterns	1041	177
Maximum number of occurrences of a pattern	40	9
Minimum number of occurrences of a pattern	1	1

get aligned with the triple. This is evident from the statistics shown in Table 4.10 as OpenIE has extracted an average of 3-4 relations from a single sentence. Table 4.11 also shows the number of patterns that are above the threshold described in Section 3.6.4.5. The threshold is set to single token matching in the alignment for subject and object, where subject and object are both comprised of single token phrases and matched with a relation which also has two arguments each having a single token. From the entire relational pattern collection, 20.89% were identified as the patterns which were above the threshold which were selected to be used as the lexicalization patterns. This is because the lexicalization pattern search process was designed to select a relational pattern only if the threshold was above 0.21 as explained in Section 3.6.4.5. Table 4.11 also summarizes the distribution of these patterns based on the grammatical gender, multiplicity and as well as the maximum and minimum occurrences of the pattern collections.

We also investigated if there was any correlation between the relation and the alignment score. The relation score determined the confidence of the extracted relation based on the training instances used to train the relation extractor, while the alignment

score determined how well the relation is aligned with the triple. However, prior to the correlation analysis, multivariate normality (MVN) of the data was examined to decide the most suitable correlation test to use with the data as correlation tests such as Pearson correlation coefficient requires data to be in a multivariate normal distribution (Field, 2010, p. 177).

Also note that the practical approach to measure the multivariate normal distribution is to measure the normality of the variables individually and to come to the conclusion that a sample is multivariate normal since univariate normality is a necessary condition for multivariate distribution (Field, 2010, p. 604). This assumption, however, is not always valid. For instance, two univariate variables may not be multivariate normal. Therefore, in this research, we used statistical tests designed to measure multivariate normality.

The multivariate normality was checked using the three statistical tests shown in Table 4.12. The table reports the results of the three multivariate normality tests, namely, Mardia's test, Henze-Zirkler test, and Royston's test. All three tests confirmed that the data is not multivariate normal. In addition, the Q-Q plot shown in Fig. 4.4 was also used to determine the multivariate normality of the data. If the data set is in a multivariate normal distribution, then the points in the Q-Q plot will approximately lie on the line $y = x$. However, it can be seen from Fig. 4.4 that there are some deviations from the straight line and this indicates possible departures from a multivariate normal distribution. The results from the MVN test and the analysis of the Q-Q plot, both form sufficient ground to declare that the data set is not multivariate normal.

Since the data is not in a multivariate normal distribution, Spearman correlation coefficient was used to calculate the correlation between the relation and the alignment score. The two tailed Spearman correlation analysis resulted in a correlation coefficient of 0.076 and a significance of 0.136. This shows that there is no significant correlation

Table 4.12 Multivariate normality analysis for Mardia’s Test, Henze-Zirkler Test, and Royston’s Test

	Mardia’s Test	Henze-Zirkler Test	Royston’s Test
Values	Estimated multivariate skewness = 0.66, chi.skew = 43.07, p.value.skew = 9.96×10^{-9} Estimated multivariate kurtosis = 5.48, z.kurtosis = -6.20, p.value.kurt = 5.53×10^{-10} chi.small.skew = 43.63, p.value.small = 7.64×10^{-9}	HZ = 28.57 p-value = 0	H = 171.35 p-value = 6.16×10^{-38}
Result	Not multivariate normal	Not multivariate normal	Not multivariate normal

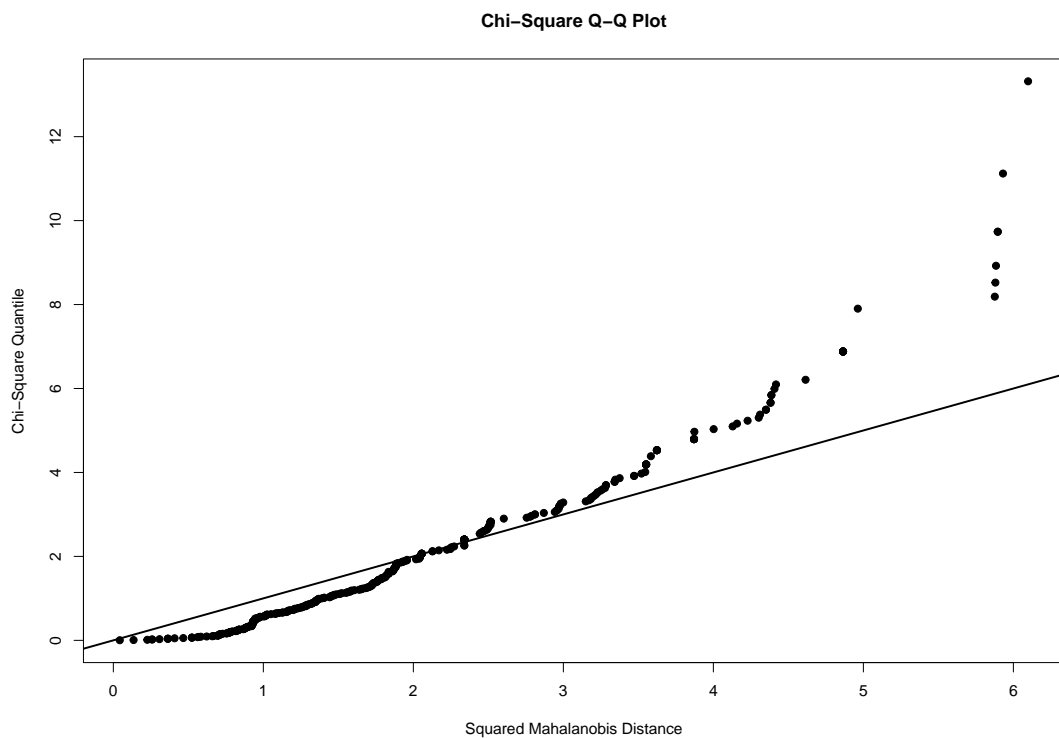


Fig. 4.4 Q-Q plot the relation score and alignment score data

Table 4.13 Examples where alignment score and relational scores have high diversity

Lexicalization pattern	Predicate	Description	Relational score	Alignment score
$\langle O?, be\ wife\ of,\ S? \rangle_L$	spouse	Used to denote the spouse of the subject	0.4031	0.8192
$\langle O?, is\ the\ wife\ of,\ S? \rangle_L$	spouse	Used to denote the spouse of the subject	0.9555	0.8192

between the relation score and the alignment score, hence it can be established that good alignments can be found in relations extracted with a low confidence score as well. Table 4.13 depicts two examples where a significant difference between the relational confidence score and alignment score can be found. The first example shows a scenario where the relational score is significantly low and the alignment score is in the vicinity of the highest value. The relational score provided by OpenIE can be lower in these scenarios where it can extract relations from sentences by processing its structure (e.g., “Michelle Obama, wife of Barack Obama”) although a direct linguistic relational phrase is not present. The second example shows an instance where the relational score is in the vicinity of the highest value and the alignment score is same as the first record. These scenarios can occur in situations where OpenIE has found direct relational phrases and the extracted relation is aligned properly with the triple. However, in this research we give the priority for the pattern with the highest relational score when multiple patterns exist with the same alignment score.

4.2.2.1.4 Property Patterns

The property patterns were based on a pattern lexicon containing five pattern templates. These templates were associated with predicates and ontology class hierarchies. Table 4.14 shows some examples of the property patterns that were used in the RealText prototype and the predicate that each pattern is associated with. Table 4.15 shows the

overall statistics of the property pattern lexicon. Table 4.15 shows that the lexicon contains 51 duplicate predicates, however, they belong to different ontology classes. The pattern search process takes the ontology class hierarchy into consideration so that if a matching record is not found it applies the RDF inference. The RDF inference can find whether a pattern exist in an upper ontology class and then apply that pattern based on the fact that the current ontology class is inherited from the upper ontology class. The contribution of the RDF inference in the property patterns will be discussed in detail in Section 4.2.2.2. The predicates in the property pattern lexicon was extracted from a total of 41 ontology classes, and ontology class hierarchies. Some examples are Place and Place→Populated Place→Settlement which contained 15 predicates that were associated with one of the five property pattern templates. We further analysed the property pattern statistics in Fig. 4.5 by categorizing them into the five different pattern templates.

Figure 4.5 shows the statistics of the property pattern lexicon categorized based on the pattern types. Some predicates in the lexicon fall under multiple ontology class hierarchies, however, the pattern templates were specified for each of the ontology class hierarchies. This is to support RDF inference which utilizes the ontology class hierarchy during the pattern search. The PP-1 has a remarkably higher representation compared to the rest of the property patterns. This is because PP-1 ($\langle\langle S? 's P?, is, O? \rangle_L$) can be applied to a number of predicates as it describes a property of an entity using the predicate as a part of the template.

4.2.2.2 Evaluation of the Lexicalization Module

This section describes the evaluation of the lexicalization module which transforms the triples into sentences using the embedded entities in the question. Some examples of the lexicalizations are shown in Table 4.16. The table shows the triple, lexicalization

Table 4.14 A set of records from the property pattern lexicon.

Ontology class hierarchy	Predicate	Type
Place → Populated Place → Region → Administrative Region	areaTotal	PP-1
Agent → Person → Artist → Comics Creator	nationality	PP-1
Agent → Person → Artist → Writer	birthName	PP-1
Agent → Person → Athlete → Racing Driver → Formula One Racer	championships	PP-2
Agent → Person → Athlete → Racing Driver → Formula One Racer	wins	PP-2
Agent → Person → Criminal	occupation	PP-3
Place → Architectural Structure → Infrastructure → Route Of Transportation → Bridge	type	PP-3
Agent → Organisation → Educational Institution → University	numberOfStudents	PP-4
Place → Populated Place → Country	largestCity	PP-4
Place	highestRegion	PP-4
Place → Populated Place → Settlement → City	isPartOf	PP-5
Place → Populated Place → Settlement	isPartOf	PP-5

pattern, the resulting lexicalization, and the lexicalization process. In addition to the lexicalizations processed by the four main pattern processors (Occupational metonyms, CFG, relational, property patterns), the lexicalizations processed by RDF inferences of these pattern processors are also shown in the table. RDF inference as explained in Section 3.6.6.1, can apply a pattern to a triple when there does not exist a pattern containing the same ontology class hierarchy as the entities in the triple, but with a higher level ontology class hierarchy. For example, the triple shown in ID-6 has the ontology

Table 4.15 Statistics of the property pattern lexicon.

Factor	Value
All property patterns	148
Unique ontology class hierarchies	41
Predicates with unique names	97
Maximum property patterns per ontology class hierarchy (Ontology class hierarchies)	15 (Place; Place→Populated Place→Settlement)
Minimum property patterns per ontology class hierarchy (Ontology class hierarchies)	1 (Agent → Person → Scientist; Agent → Person → Criminal, etc.)

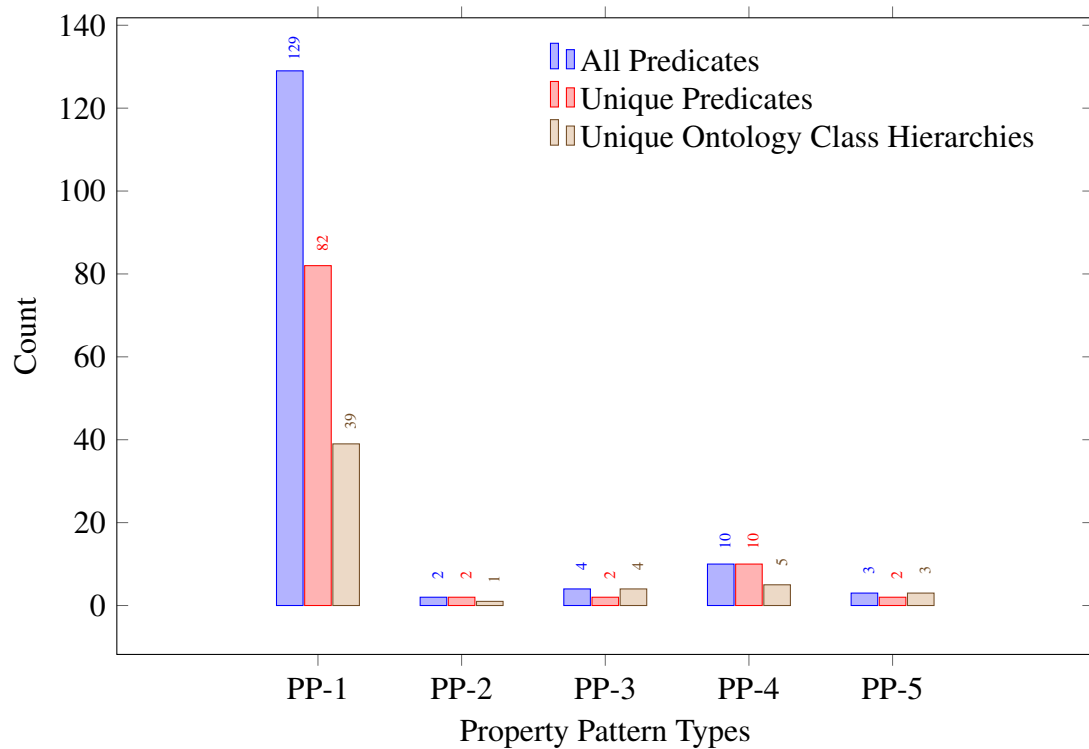


Fig. 4.5 Statistics of the property pattern lexicon categorized based on the pattern type

class hierarchy $\text{Agent} \rightarrow \text{Person} \rightarrow \text{Athlete} \rightarrow \text{RacingDriver} \rightarrow \text{FormulaOneRacer}$ and the ontology class hierarchy of the pattern is $\text{Agent} \rightarrow \text{Person}$. Although ontology class hierarchies are different, the former is an inherited hierarchy of the latter, and therefore the lexicalizations of the properties that are inherited from the *Person* ontology class can also be applied to *Formula One Racer*. For ease of reference, the rest of the section uses terms “direct pattern” and “inference based patterns” to differentiate the patterns that are applied directly (e.g., relational pattern) and patterns that are applied using the RDF inference (e.g., relational pattern applied using RDF inference).

We first analysed the pattern type distribution among the 52 questions in the test dataset. Figure 4.6 depicts the result of this analysis for the four main pattern types. This graph illustrates the inference based patterns as part of the main pattern type. Later in this section, we analyse how inference based patterns contributed and compare them with the direct pattern application.

The graph in Fig. 4.6 shows that for 32 question (61.53% of the dataset), the relational patterns were able to lexicalize more triples than the other three patterns. For the test set, the lexicalization patterns were again able to lexicalize a large proportion (53.16%) of the questions. In addition, relational patterns contributed to the lexicalization for all of the 52 questions in the test set,. The other three patterns — occupational metonym, CFG, and property patterns — contributed towards 21, 4, and 47 questions respectively. The high success rate of the relational patterns is due to a number of reasons. Firstly, the relational pattern processing module is designed to find patterns from unstructured text and does not limit its capability to a lexicon. The relational extraction employed in the pattern processing stage can extract a wide range of relations from the unstructured text which can be aligned with triples. The occupational metonym patterns which used the occupational metonym lexicon can only lexicalize the selected amount of triples as the predicate of the triple should be an occupational metonym

Table 4.16 Example lexicalizations from the test data. Note that subject expression is replaced with actual subject value in the resulting lexicalization. However, this substitution will occur after the referring expression generation in the framework.

ID	Triple and Lexicalization pattern	Resulting lexicalization	Lexicalization process
1	<p>⟨<i>Battlestar Galactica, executive Producer, Ronald D. Moore</i>⟩_T ⟨<i>S?, was produced by, O?</i>⟩_L</p>	<p>⟨<i>Battlestar Galactica, was produced by, Ronald D. Moore</i>⟩_{LT}</p>	Occupational Metonymy
2	<p>⟨<i>John F. Kennedy, successor, Lyndon B. Johnson</i>⟩_T ⟨<i>O?, succeeded, S?</i>⟩_L</p>	<p>⟨<i>John F. Kennedy, succeeded, Lyndon B. Johnson</i>⟩_{LT}</p>	Occ. Metonymy - Inference
3	<p>⟨<i>Microsoft, founded By, Bill Gates</i>⟩_T ⟨<i>S?, is founded by, O?</i>⟩_L</p>	<p>⟨<i>Microsoft, is founded by, Bill Gates</i>⟩_{LT}</p>	CFG
4	<p>⟨<i>Angela Merkel, birth Date, 1954-07-17</i>⟩_T ⟨<i>S?, was born on, O?</i>⟩_L</p>	<p>⟨<i>Angela Merkel, was born on, July 17, 1954</i>⟩_{LT}</p>	Relational
5	<p>⟨<i>Microsoft, location City, Redmond</i>⟩_T ⟨<i>S?, is located in, O?</i>⟩_L</p>	<p>⟨<i>Microsoft, is located in, Redmond</i>⟩_{LT}</p>	Relational
6	<p>⟨<i>Rubens Barrichello, birth Place, Sao Paulo</i>⟩_T ⟨<i>S?, was born in, O?</i>⟩_L</p>	<p>⟨<i>Rubens Barrichello, was born in, Sao Paulo</i>⟩_{LT}</p>	Relational Inference
7	<p>⟨<i>Lisbon, highest Region, Benfica (Lisbon)</i>⟩_T ⟨<i>highest region in S?, is, O?</i>⟩_L</p>	<p>⟨<i>Highest region in Lisbon, is, Benfica (Lisbon)</i>⟩_{LT}</p>	Property Pattern
8	<p>⟨<i>Rembrandt, nationality, Dutch</i>⟩_T ⟨<i>S?'s nationality, was, O?</i>⟩_L</p>	<p>⟨<i>Rembrandt's nationality, was, Dutch</i>⟩_{LT}</p>	Property Pattern - Inference

and ontology class hierarchy should match with or must be inherited from one which is already recorded in the lexicon. Similarly, CFG also contained the restriction that predicate should be a verb and has a verb frame that match with which is defined by grammar rule (\mathcal{G}). These restrictions have led the low number of lexicalizations of occupational metonym and CFG patterns. However, property patterns which work on pattern templates can be applied in a wide number of predicates and therefore contributed for lexicalization higher than CFG and occupational metonym patterns. Furthermore, the graph in Fig. 4.6 shows that the pattern type distribution does not have consistent performance among the questions.

The results in Fig. 4.6 can be further analysed for the contribution of RDF inference together with the distribution of the 5 property patterns among the lexicalizations for the test data question set. Firstly, Fig. 4.7 shows the inference based pattern distribution. According to the figure, occupational metonym inference patterns, relational inference patterns, and property pattern inference have the contributions of 46.78%, 21.09%, and 12.29% of lexicalizations respectively, compared to the total number of lexicalizations done by corresponding patterns. These results are in accord with the design of the RDF inference as inference was considered only if the direct pattern search failed to select a pattern.

Figure 4.8 illustrates the property pattern distribution based on the type of the pattern. The PP-1 has covered most of the triples compared to the rest of the property pattern types. This is mainly because the PP-1, $\langle S?'s P?, is, O? \rangle_L$, can be applied to a number of predicates as this pattern shows an attribute or a property of the subject entity. Figure 4.8 also shows that except for the PP-1, all of the other property pattern types did not contribute via inference.

The lexicalization module also does realizations in both the active person and gender realization forms, during the pattern application stage to enhance the lexicalization into

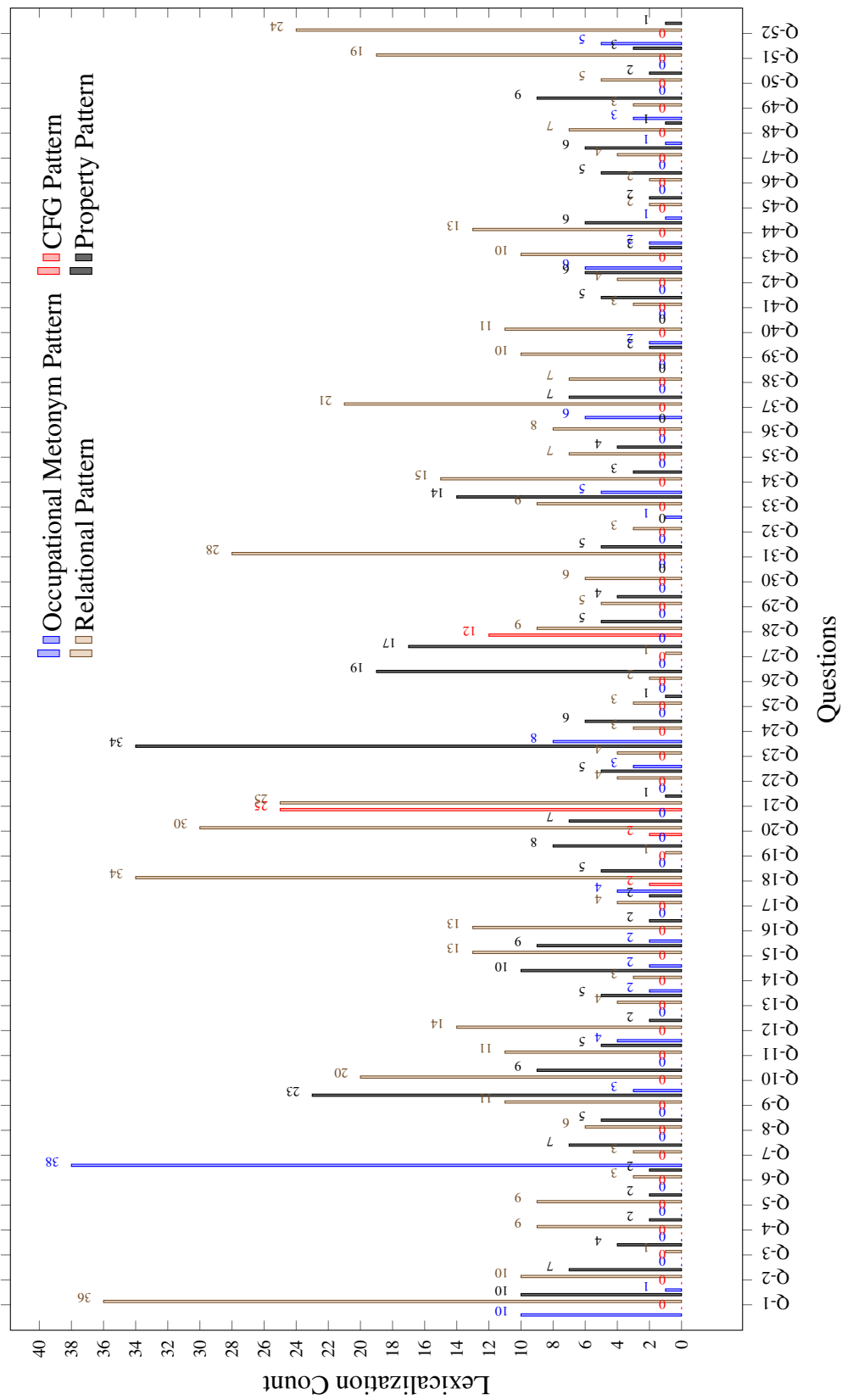


Fig. 4.6 Question wise lexicalization pattern type distribution

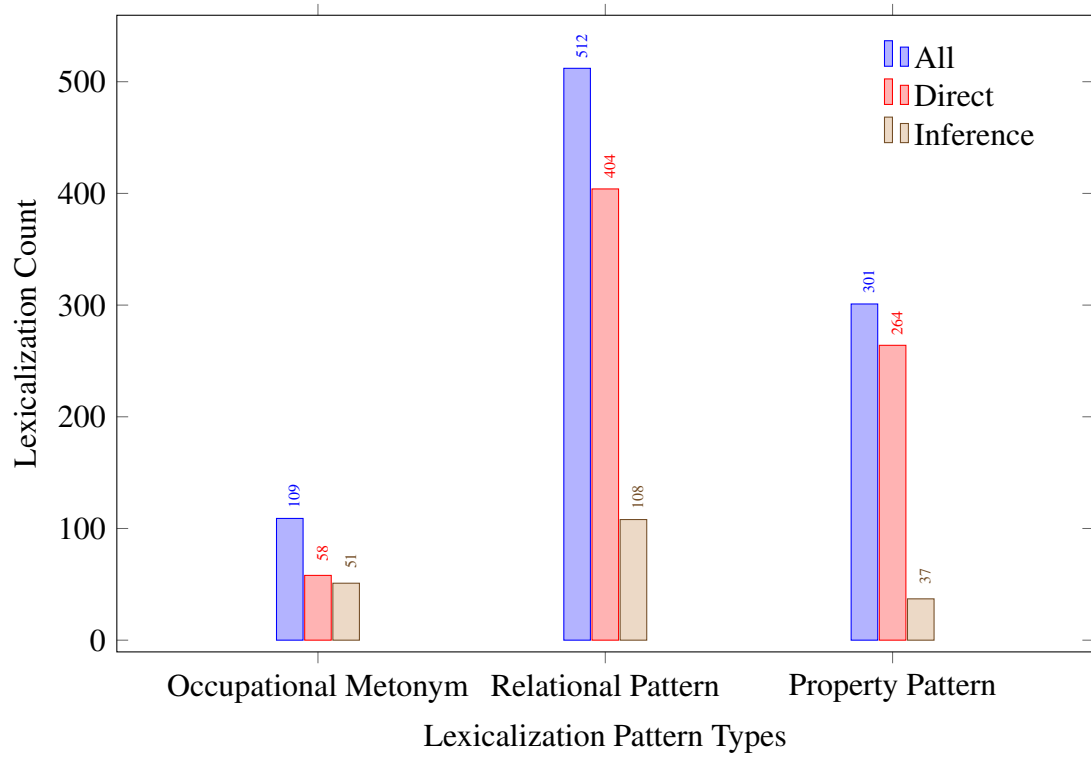


Fig. 4.7 Inference based pattern distribution

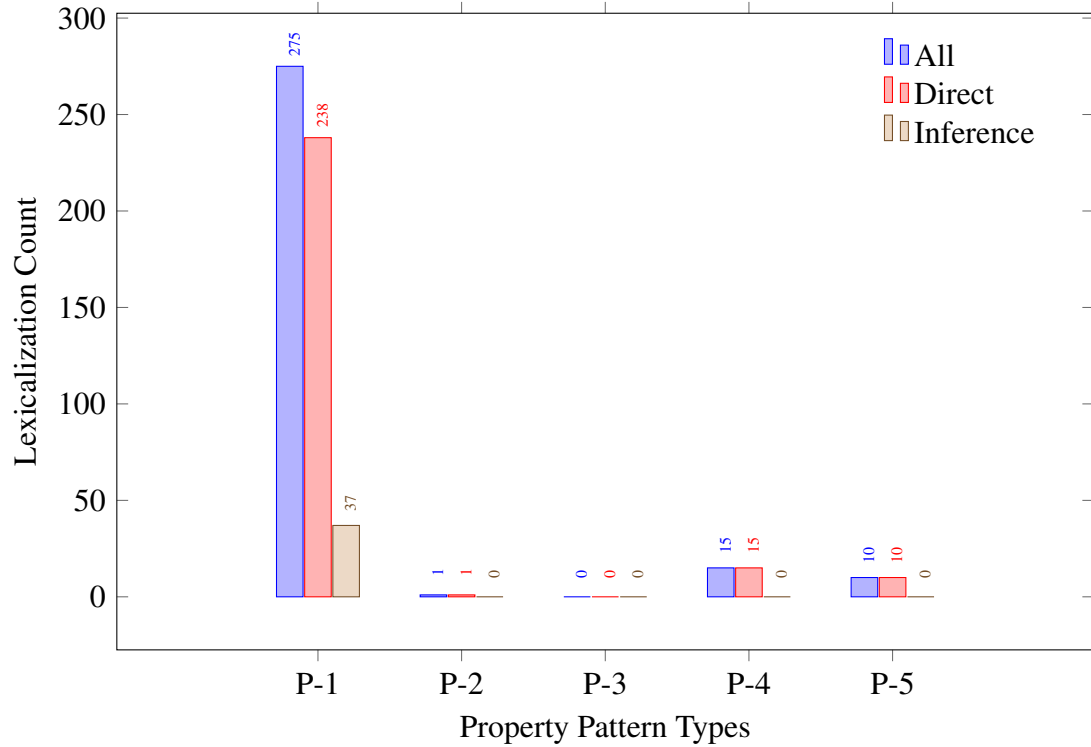


Fig. 4.8 Property pattern type wise distribution

a form which is even closer to a human generated form. During the experiment with the test dataset, lexicalization framework has carried out 13 gender realizations and 77 person active realizations. Table 4.17 reports some of the realizations carried out during the lexicalization pattern application stage.

Figure 4.9 depicts the accuracy of the lexicalization patterns distributed among the question set consisting of the 52 questions in the test set. The graph shows the number of lexicalizations, and the number of lexicalizations which are both syntactically and semantically correct. We evaluated syntactic and semantic accuracy of lexicalizations separately and then analysed how many of them are correct in both terms. The main reason behind evaluating the syntactic and semantic accuracy separately and to later analyse how many of them are correct in both terms is that it can reveal both the syntactic and semantic perspectives of the lexicalization process. According to the graph, in 26 questions (50% of the entire question set), all lexicalizations that were applied for the triples were syntactically and semantically correct. When considering the whole collection of triples, 887 triples out of the 963 lexicalized triples were associated with both syntactically and semantically accurate lexicalization patterns. This yields to 92.10% accuracy on the lexicalized triple collection. This emphasizes that once the lexicalization pattern is applied to a triple, they are highly accurate when considering both syntactic and semantic aspects. Even when considering the whole triple collection (including which are not associated with lexicalization patterns) of 1421 triples, the both syntactically and semantically correct lexicalizations result in 62.42% accuracy.

Compared to the Walter et al.'s (2013) approach which achieved 37% in full automatic mode and 76% in semi-automatic mode, RealText lexicalization has achieved an accuracy of 62.42% in the full automatic mode which was defined as execution of the whole pipeline without any human intervention. Walter et al.'s approach reached an accuracy of 76% (with a 105.4% boost compared to Walter et al.'s framework full

Table 4.17 Examples of realizations

Triple	Lexicalization pattern	Lexicalized triple	Lexicalized triple (realized)	Realization form
$\langle \text{Michael Jackson, parent, Katherine Jackson} \rangle_T$	$\langle S? 's, \text{father was, } O? \rangle_L$	$\langle \text{Michael Jackson 's, father was, Katherine Jackson} \rangle_{LT}$	$\langle \text{Michael Jackson 's, mother was, Katherine Jackson} \rangle_{LT}$	Gender
$\langle \text{Margaret Thatcher, child, Carol Thatcher} \rangle_T$	$\langle S? 's, \text{son was, } O? \rangle_L$	$\langle \text{Margaret Thatcher 's, son was, Carol Thatcher} \rangle_{LT}$	$\langle \text{Margaret Thatcher 's, daughter was, Carol Thatcher} \rangle_{LT}$	Gender
$\langle \text{Lyndon B. Johnson, party, Democratic Party} \rangle_T$	$\langle S?, \text{is a member of, } O? \rangle_L$	$\langle \text{Lyndon B. Johnson, is a member of, Democratic Party} \rangle_{LT}$	$\langle \text{Lyndon B. Johnson, was a member of, Democratic Party} \rangle_{LT}$	Person active
$\langle \text{Lyndon B. Johnson, child, Lynda Bird Johnson Robb} \rangle_T$	$\langle S? 's, \text{daughter is, } O? \rangle_L$	$\langle \text{Lyndon B. Johnson 's, daughter is, Lynda Bird Johnson Robb} \rangle_{LT}$	$\langle \text{Lyndon B. Johnson 's, daughter was, Lynda Bird Johnson Robb} \rangle_{LT}$	Person active

automatic mode) only through the human intervention. We did not carry out any human intervention experiments in lexicalization, since in a scalable environment with a massive amount of triples, any human intervention would not be feasible.

However, in certain scenarios, some lexicalization applied were semantically incorrect, although they are syntactically accurate. The main reasons for this semantic inaccuracy can be classified into two categories; content issues and applicability issues. Table 4.18 shows four examples of content issues in the Linked Data resource which lead the lexicalization process to generate inaccurate lexicalizations. These scenarios clearly show that the incomplete object values in the triples have caused both syntactic and semantically incorrect lexicalizations. Although these triples are expected to be accurate and complete, completion of this data was not part of this research, hence the lexicalization was done without any human intervention. There are also errors caused due to the application of incorrect lexicalizations. The last record in the Table 4.18 is both syntactically and semantically correct lexicalization if it is considered individually. However, the applied lexicalization does not describe the semantics associated with the triple.

The lexicalization evaluation was further analysed to determine the accuracy of the pattern wise distribution. Figure 4.10 and Fig. 4.11 depict the accuracy of the lexicalization categorized into main pattern types and inference level categorization respectively. As shown in Fig. 4.10 occupational metonym, CFG, and property patterns have achieved the full accuracy while relational patterns have achieved 85.15%. The incorrect patterns were mainly due to the content and applicability issues mentioned in Table 4.18. As occupational metonyms, CFG and property patterns are based on lexicons, the possibility of an error occurring in applicability is very low and in this particular experiment there were no errors reported. This is because lexicon based pattern modules utilized the pattern templates and these templates were applied under

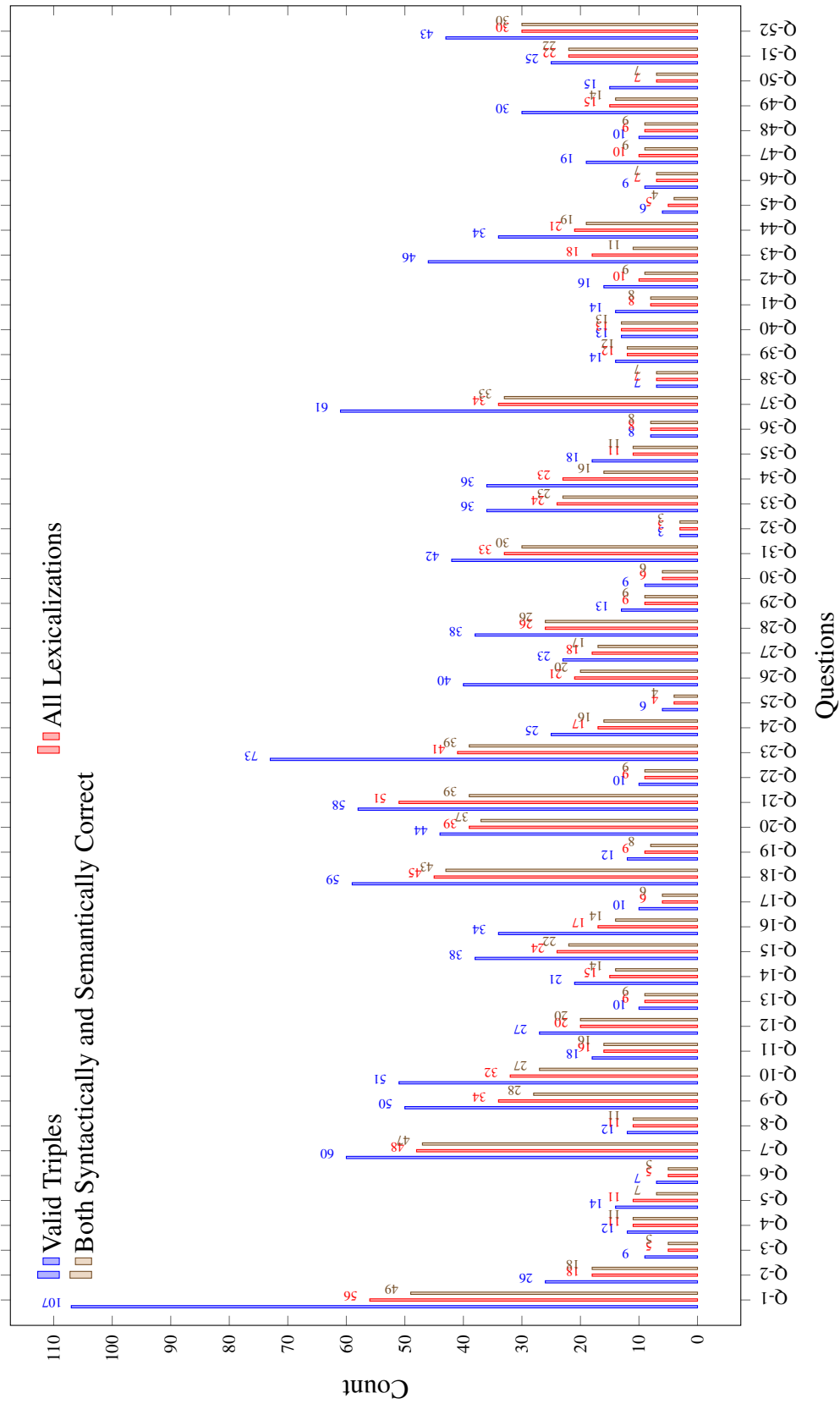


Fig. 4.9 Question wise lexicalization accuracy

Table 4.18 Some examples of reasons behind inaccurate lexicalization

Triple and lexicalization pattern	Issue description	Reason
<p><i>⟨Skype Technologies, product, Voice over IP⟩_T</i> <i>⟨S?, produces, O?⟩_L</i></p>	<p>The Voice Over IP (VOIP) is not a product that can be produced. The main idea of having this triple is to explain that Skype Technologies uses the VOIP. However, the content does not match with the underlying semantics. This makes the lexicalization pattern syntactically correct, but semantically incorrect.</p>	Content issue
<p><i>⟨Microsoft, subsidiary, List of mergers and acquisitions by Microsoft⟩_T</i> <i>⟨S?, acquired, O?⟩_L</i></p>	<p>This is also a similar scenario to the above. The lexicalization pattern is intended for single acquisition as expected by the triple. However, the issue is in the content of this triple which describes a list of acquisitions. This has caused both syntactic and semantic errors in lexicalization.</p>	Content issue
<p><i>⟨Intel, product, Bluetooth⟩_T</i> <i>⟨S?, produces, O?⟩_L</i></p>	<p>The issue here is that the content is partial and therefore with the syntactically correct pattern, the lexicalization does not express the core idea of the triple.</p>	Content issue
<p><i>⟨Barack Obama, office, from the 13th District⟩_T</i> <i>⟨S?, was, the O?⟩_L</i></p>	<p>This is similar to the previous scenarios where content is expressed partially leading to the incorrect semantics even with the correct lexicalization pattern.</p>	Content issue
<p><i>⟨London, country, United Kingdom⟩_T</i> <i>⟨S?, is the capital of, the O?⟩_L</i></p>	<p>In this example, the triple is intended to describe that London is a city in United Kingdom. In fact the lexicalization itself is both semantically and syntactically correct, if we consider them without the original triple being lexicalized. However, the lexicalization does not express the actual semantic of the triple.</p>	Applicability issue

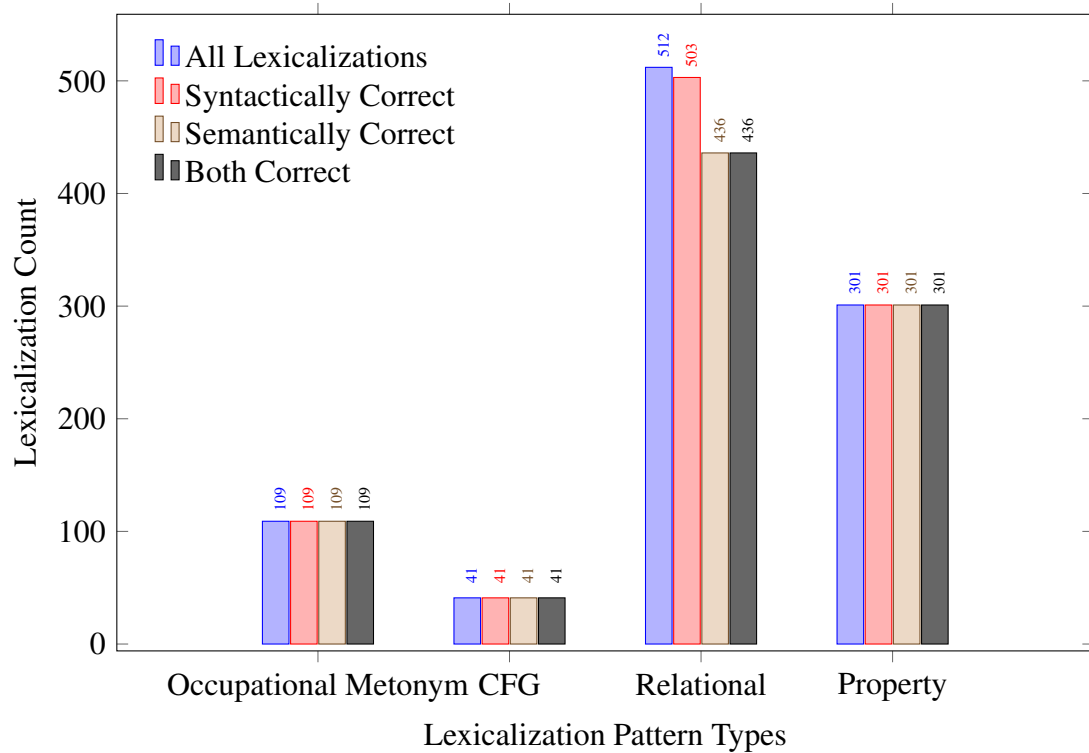


Fig. 4.10 Type wise lexicalization pattern accuracy

restrictions. On the other hand, relational pattern processing modules extracted the lexicalization patterns by aligning triples with relations extracted from the unstructured text, which has a possibility of errors compared to the lexicon based approach. However, even in lexicon based approaches, there is a possibility of an error occurring in content (e.g., partial content) as seen in Table 4.18. In our current experiment all content issues were associated with the relational pattern based lexicalizations which have the highest representation in the lexicalization.

Figure 4.11 reports the accuracy distribution of the inference based relational and property patterns. As there were no errors reported in property patterns, those which were applied based on the inference were all accurate. However, the accuracy of the inference based relational pattern was reported as 96.29%, where 4 patterns were identified as inaccurate from the 108 inference based patterns.

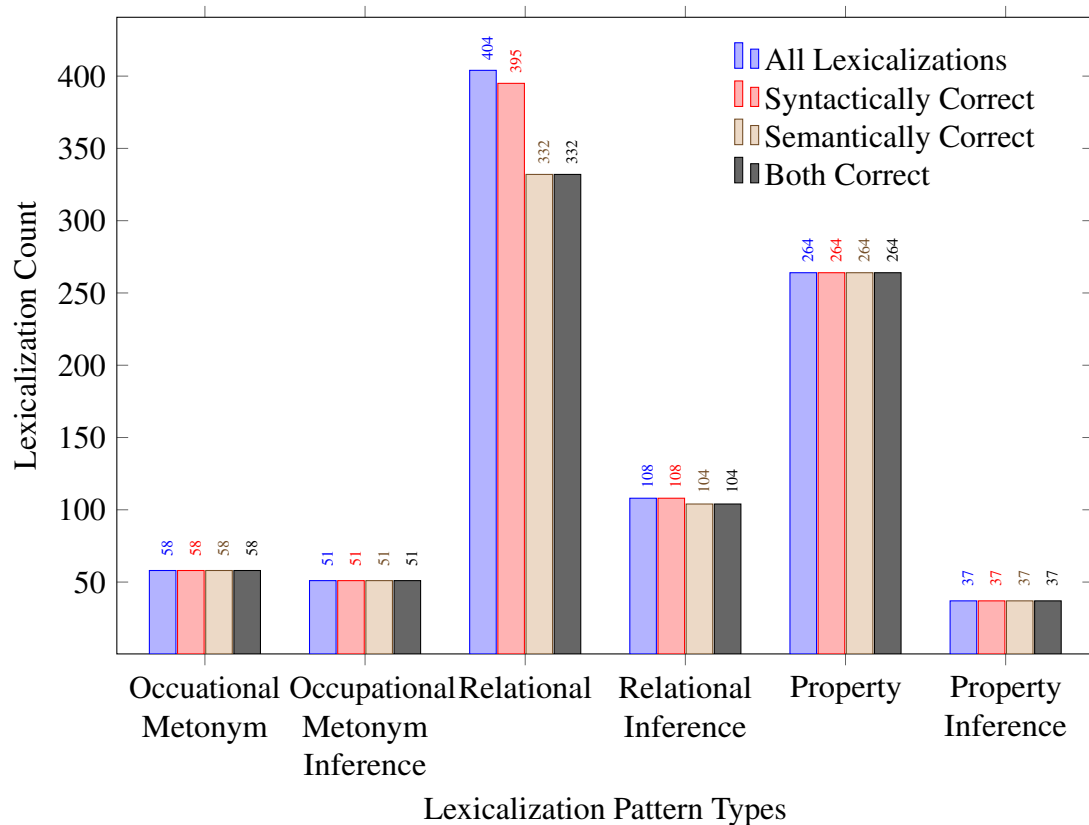


Fig. 4.11 Inference level pattern accuracy

4.2.3 Aggregation

The aggregation of the RealText framework was carried out using eight rules described in Chapter 3. This section describes how these rules contributed to the aggregations in the test dataset comprised of 52 questions with some examples taken from the test dataset aggregations. Firstly, Fig. 4.12 shows the number of aggregations achieved by the individual rules. From the eight rules specified, Rule-1 contributed mostly to aggregate lexicalizations which constitutes 45.02% of the total aggregations. This is mainly because Rule-1 targets aggregating triples in which the subject and predicates are similar but the object is different. Table 4.19 shows a selection of examples of triple aggregations from the question dataset.

Table 4.19 Example aggregations retrieved from the test dataset

Triples	Lexicalization Patterns	Aggregated Result	Aggregation Rule
<p>⟨Neil Gaiman, influenced By, Michael Moorcock⟩_T</p> <p>⟨Neil Gaiman, influenced By, Gene Wolfe⟩_T</p> <p>⟨Neil Gaiman, influenced By, Jack Vance⟩_T</p>	<p>⟨S?, is influenced by, O?⟩_L</p> <p>⟨S?, is influenced by, O?⟩_L</p> <p>⟨S?, is influenced by, O?⟩_L</p>	<p>⟨S? is influenced by Michael Moorcock, Gene Wolfe, and Jack Vance⟩_{AG}</p>	Rule-1
<p>⟨Harold and Maude, writer, Colin Higgins⟩_T</p> <p>⟨Harold and Maude, producer, Colin Higgins⟩_T</p>	<p>⟨S?, was written by, O?⟩_L</p> <p>⟨S?, was produced by, O?⟩_L</p>	<p>⟨S? was written, and produced by Colin Higgins⟩_{AG}</p>	Rule-2
<p>⟨Klaus Wowereit, birth date, 1953-09-30⟩_T</p> <p>⟨Klaus Wowereit, birth place, West Germany⟩_T</p>	<p>⟨S?, was born on, O?⟩_L</p> <p>⟨S?, was born in, O?⟩_L</p>	<p>⟨S? was born on 1953-09-30 in West Germany⟩_{AG}</p>	Rule-3
<p>⟨K2, prominence, 4017⟩_T</p> <p>⟨K2, elevation, 8611⟩_T</p>	<p>⟨S?'s prominence, is, O?⟩_L</p> <p>⟨S?'s elevation, is, O?⟩_L</p>	<p>⟨S?'s prominence, and elevation are respectively 4017.0 m, and 8611.0 m⟩_{AG}</p>	Rule-5
<p>⟨Michael Jackson, parent, Joe Jackson⟩_T</p> <p>⟨Michael Jackson, parent, Katherine Jackson⟩_T</p>	<p>⟨S?'s, father is, O?⟩_L</p> <p>⟨S?'s, mother is, O?⟩_L</p>	<p>⟨S?'s parents are Joe Jackson , and Katherine Jackson⟩_{AG}</p>	Rule-7

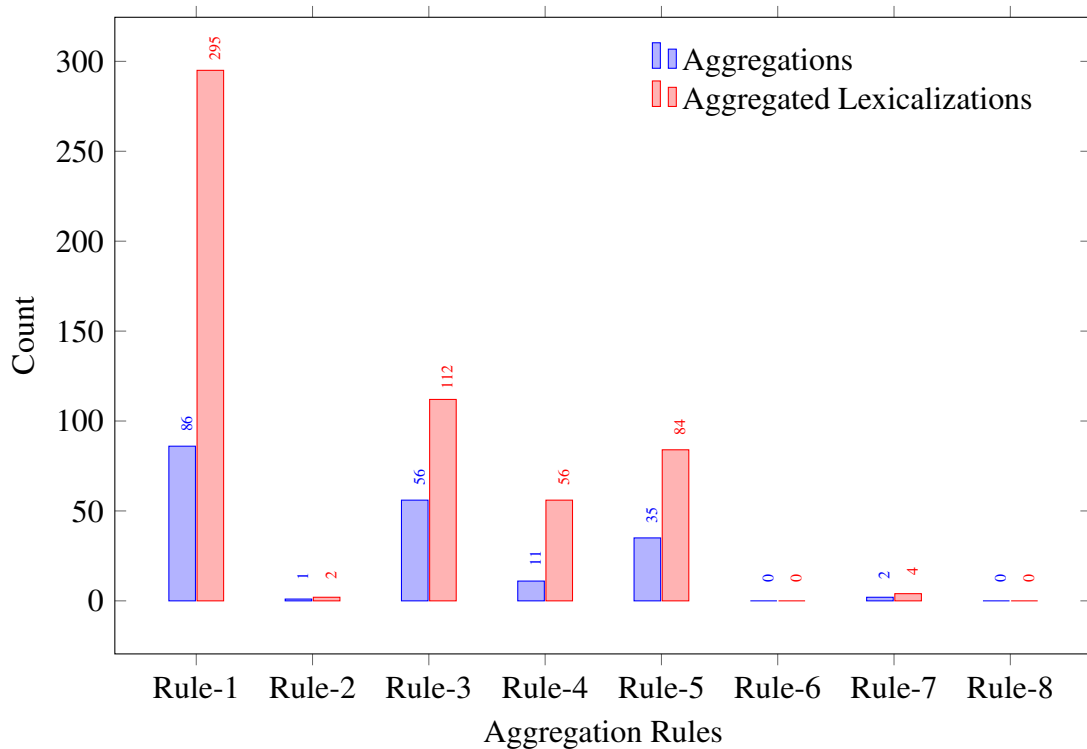


Fig. 4.12 Distribution of aggregations based on the rule (only unique ones are recorded)

The aggregation module also integrates a word pluralization process to syntactically correct sentences. This process has pluralized 33 tokens for the test dataset. In addition, 6 of them were treated as “pluralizer exceptions”, where we have introduced new plural tokens based on multiplicity of the token (e.g., son + daughter \Rightarrow children). In the number of generated answers these were applied as expected and it shows the viability and importance of such a token pluralization strategy which has direct benefits on the readability.

Figure 4.13 illustrates the percentage of contribution of each aggregation rule for the 52 questions in the test dataset. It is apparent from the graph that no aggregations have been carried out in six of the questions. This is because the current rule set did not identify any possible aggregations in the lexicalized triples of these questions. Furthermore, the significant contribution of Rule-1 is further confirmed from the graph where 63.46% of the dataset (33 questions) involves Rule-1 aggregations. It is also

evident from the graph that multiple rules contribute to accurately aggregate lexicalized triples for the individual questions. For example, aggregations in Q-15 were carried out by 4 different rules. Such an aggregation is crucial in machine generated text as it increases the language richness akin to human generated texts.

We also analysed the ratio between lexicalized triples and the aggregated ones which is shown in Fig. 4.14. According to the results only 34.15% of the question dataset (18 questions) has less than 50% of the lexicalized triples which were aggregated. This includes the six questions with no aggregation. Based on both Fig. 4.13 and Fig. 4.14, it is clear that various aggregations were performed in most of the questions using different combinations of the rules. However, it should be noted that at this stage of the pipeline, the lexicalized and aggregated triples do not have their subject resolved (subject is denoted as S?). The next section describes the evaluation of the process of resolving these unresolved subject entities with referring expressions.

4.2.4 Referring Expression Generation

This section presents the statistics related to the referring expression generation where subjects of the lexicalized and aggregated triples were given referring expressions to improve the naturalness of the generated text. A naive language generation without referring expressions will lead the entity names being repeated, thus making it harder for the hearer to follow the discourse and link different pieces of information related to a single entity. With the use of the referring expressions, the hearer can easily follow an entity, as an entity is given an expression which is easy to remember.

The referring expressions were applied from several categories, namely, personal pronouns, possessives, entity name variations, and words from various ontology classes that are used as referents. Table 4.20 shows the statistics related to the referring expression generation. It is clear from the statistics that personal pronouns are more prevalent

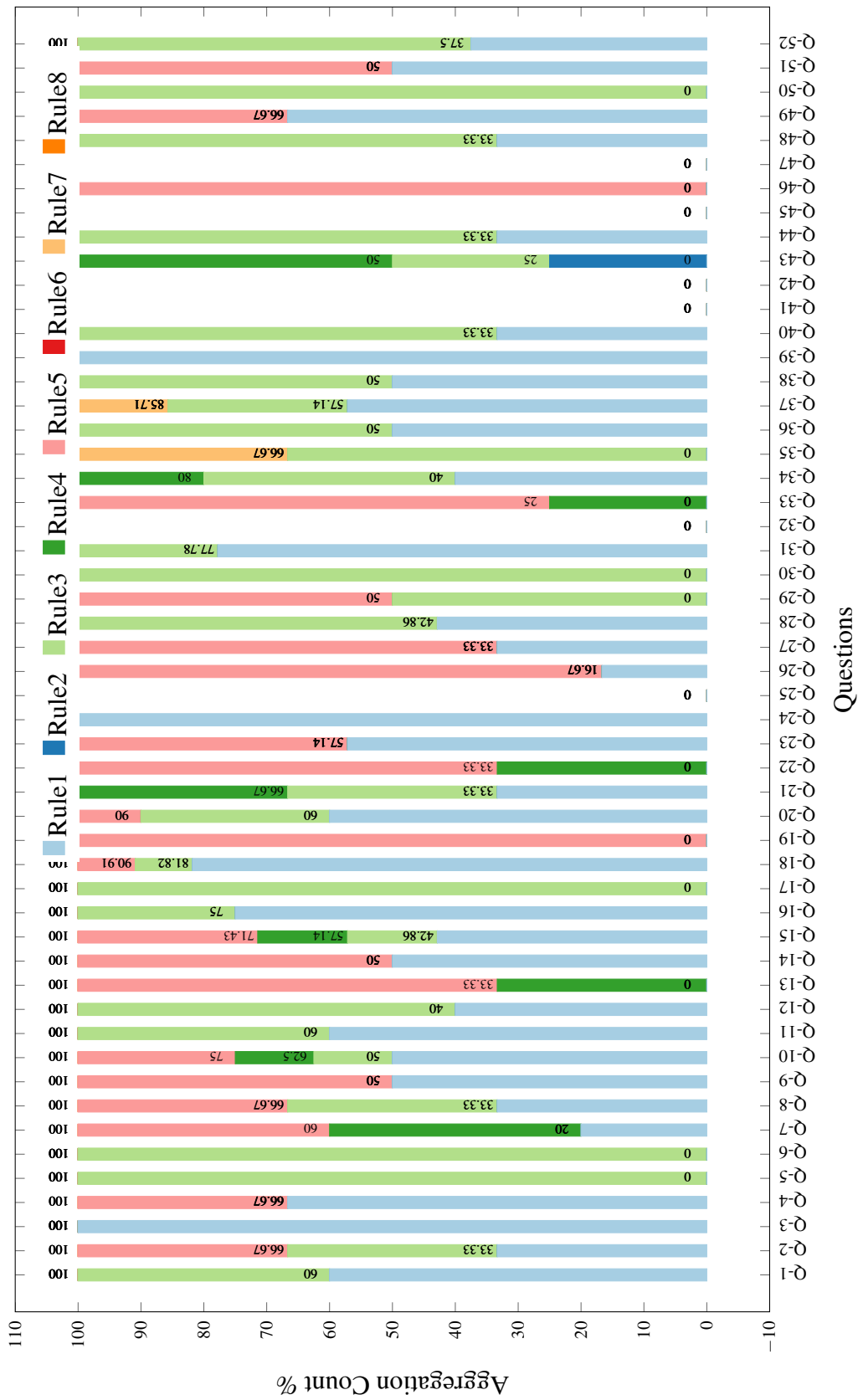


Fig. 4.13 Aggregation rule distribution per question

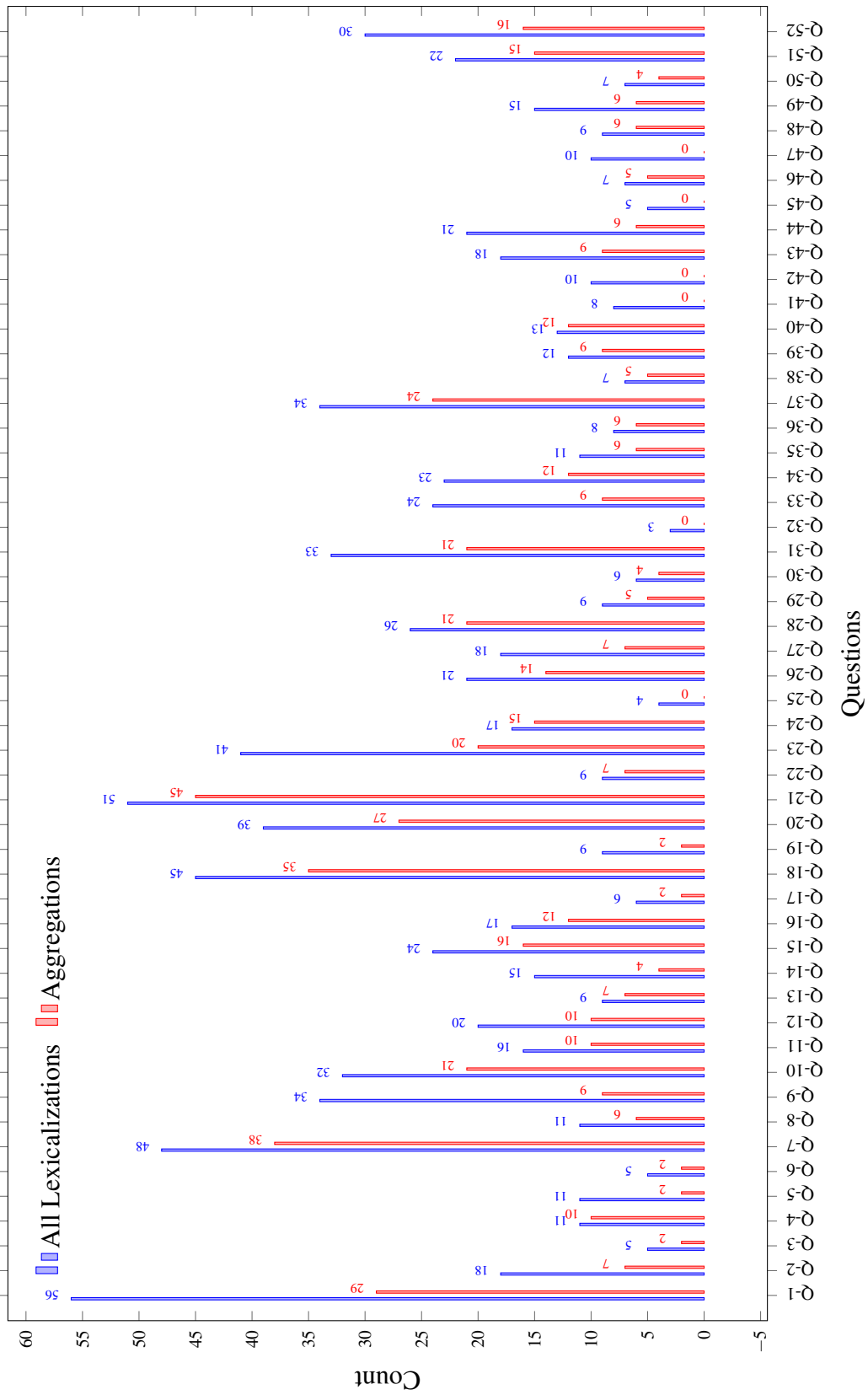


Fig. 4.14 All lexicalization versus the aggregated lexicalizations

Table 4.20 Referring expression generation statistics

Factor	Value
Number of personal pronouns for persons entities (e.g., he, she)	120
Number of possessives for person entities (e.g., his, her)	83
Number of personal pronouns for things (e.g., it)	117
Number of possessives for things (e.g., its)	100
Number of referring expressions with entity name	45
Number of referring expressions with entity possessions	19
Number of referring expressions with ontology class	7
Number of referring expressions with ontology class possessions	0

than possessives. Since the referring expressions were pulled from a pool of referring expressions while managing the language variety, we used adjacent occurrences of same referring expression a maximum of two times. Furthermore, ontology classes were used only in the second sentence and not repeated in the same narrative. The complete analysis of all the referring expressions in the test dataset showed that there is no ambiguous use of referring expressions.

Table 4.21 shows some example referring expressions generated by the framework. The framework used the gender database to determine the personal pronoun or possessives which need to be used to refer to an entity. In addition, the table also shows the variations of entity names (i.e., the first name of a person) used as the referring expression. The ontology class of the entity was used only in the second appearance of the entity name as repeated use can make it harder for the hearer to link the referring expression to the entity as ontology classes are generic terms (e.g., company, organization). However, there were situations where multiple entities appeared that can be referred using a common expression. For example, the question “who was the successor of John F. Kennedy?” has the answer as “Lyndon B. Johnson”, where both the entities are male and can be referred by using personal pronouns *he*, *his*, or *him*. If the descriptions related to these two entities appear as a single paragraph, then the

reader will get confused when linking referring expression to the entity. This issue was resolved by the framework by introducing subject based clustering in aggregation, leading the framework to generate multiple paragraphs for different entities. Such separation of description to avoid the confusion between the referring expression is another use of the Centring theory as discussed in Section 3.7 and Section 3.8.

Table 4.21 Example referring expressions generated for the test dataset

Referring expressions	Type of the expression
Lyndon B. Johnson was an office holder. He was born on August 26, 1908 in Texas.	Personal pronouns
Vrije Universiteit is a university. It is located in Amsterdam	Personal pronouns
Neil Gaiman is a writer. His birth name is Neil Richard Gaiman	Possessives
Lisbon is capital of Portugal. Its elevation, minimum elevation, and maximum elevation are respectively 2.0 m, 0 cm, and 199.0 m	Possessives
Benjamin Franklin was a governor. Benjamin died on April 17, 1790 in Philadelphia.	Entity name variations
Michael Jackson was born on August 29, 1958 in Gary, Indiana. Michael's residence was Neverland Ranch.	Entity name variations
Skype Technologies is a company. The company is located in Luxembourg.	Ontology classes
Microsoft is a company. The company was founded on April 04, 1975 in 1975.	Ontology classes

4.2.5 Structure Realization

Structure realization presented the generated answers in five different formats: SSML, HTML, \LaTeX , ODF, and RDF. Listing 4.1 and Listing 4.2 show examples from the test

dataset where the answer was presented in SSML and RDF forms. As shown in the Listing 4.1¹, SSML based structure realization annotated the answer with tags which help the speech synthesizer to transform the generated answer into a voice without ambiguities. For example, a year such as “1953” can be interpreted as a number by the speech synthesizer, however, it will be interpreted without any error when annotated with correct format such as `<say-as interpret-as=“date”>1953</say-as>`. Additionally, structure realization also presented the generated answer as a RDF triple form which is the same form initially used to retrieve information to generate the answers. As shown in Listing 4.2, the generated answer is a human readable RDF answer which originated from the machine readable RDF triples.

4.3 Human Evaluation Results

As argued in Section 2.5, human evaluation still stands as the most appropriate and accurate evaluation mechanism to rate machine generated natural language answers. This section describes the human evaluation performed in order to evaluate the answers generated from the RealText framework. The following sections describe the process of evaluation, results, and a detailed analysis based on the acquired results.

4.3.1 Evaluation Process

The evaluation focused on three major criteria described below with the definition and the scope.

- **Readability and clarity (hereinafter referred to as Readability):** This focuses on evaluating the language quality of the generated answers. Readability measures

¹The audio version of this answer generated by a speech synthesizer can be accessed from: <http://www.rivinduperera.com/projects/realtext/>

```
1 <paragraph>
2 <sentence><emphasis>Jack Kirby and Joe Simon</emphasis> created
   the comic Captain America</sentence>
3 </paragraph>
4
5 <paragraph>
6 <sentence>Captain America is a comics character. It was created by
   Jack Kirby, and Joe Simon.</sentence>
7 <sentence>Jack Kirby was a comics creator. He was an American book
   artist.</sentence>
8 <sentence>He was born on <say-as interpret-as="date">August 28,
   1917</say-as> in New York City.</sentence>
9 <sentence>Jack died on <say-as interpret-as="date">February 06,
   1994</say-as> in Thousand Oaks, California.</sentence>
10 </paragraph>
11
12 <paragraph>
13 <sentence>Joe Simon was a comics creator.</sentence>
14 <sentence>He was born on <say-as interpret-as="date">October 11,
   1913</say-as> in New York.</sentence>
15 <sentence>He won the Eisner Award, and the Inkpot Award.</sentence
   >
16 <sentence>Joe died on <say-as interpret-as="date">December 14,
   2011</say-as> in New York City.</sentence>
17 </paragraph>
18
```

Listing 4.1 Answer generated for the question “Who created the comic Captain America?” which is presented in SSML form

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xmlns:owl="http://www.w3.org/2002/07/owl#"
6   xmlns:rtp="http://realtext.org/property/"
7   xmlns:foaf="http://xmlns.com/foaf/0.1/"
8   xmlns:prov="http://www.w3.org/ns/prov#" >
9   <rtp:answerSentence xml:lang="en">
10    Jack Kirby and Joe Simon created the comic Captain America
11  </rtp:answerSentence>
12  <rtp:entityLabel xml:lang="en">
13    Captain America
14  </rtp:entityLabel>
15  <rtp:entityLabel xml:lang="en">
16    Jack Kirby
17  </rtp:entityLabel>
18  <rtp:entityLabel xml:lang="en">
19    Joe Simon
20  </rtp:entityLabel>
21  <rtp:entityDescription xml:lang="en">
22    Captain America is a comics character. It was created by Jack
23      Kirby, and Joe Simon.
24  </rtp:entityDescription>
25  <rtp:entityDescription xml:lang="en">
26    Jack Kirby was a comics creator. He was an American book artist.
27      He was born on August 28, 1917 in New York City. Jack died on
28      February 06, 1994 in Thousand Oaks, California.
29  </rtp:entityDescription>
30  <rtp:entityDescription xml:lang="en">
31    Joe Simon was a comics creator. He was born on October 11, 1913 in
32      New York. He won the Eisner Award, and the Inkpot Award. Joe
33      died on December 14, 2011 in New York City.
34  </rtp:entityDescription>
35 </rdf:RDF>
```

Listing 4.2 Answer generated for the question “Who created the comic Captain America?” which is presented in RDF form

the level of which the generated answers are linguistically correct so the answers can be understood by the readers.

- Accuracy and appropriateness (hereinafter referred to as Accuracy): This factor focuses on the content quality of the generated answers. Accuracy checks whether triples are correctly expressed in natural language.
- Informativeness: This criteria evaluates the informativeness of the answers. While accuracy only focuses on the quality of the generated answers, informativeness checks whether answers present enough information for the reader to get a clear understanding of the entities being discussed.

The use of readability and accuracy criteria in this research is influenced by the detailed survey carried out by Reiter and Belz (2009) on evaluating machine generated text. Although readability and accuracy cover the essential features of the generated text, our evaluation also needed ranking of informativeness. This is mainly to assess whether generated answers could be rated on the amount of information content in the generated text.

The evaluation was carried out as a survey² where we expected at least 10 valid responses to evaluate the answers with an acceptable agreement . The survey was given to 20 volunteer participants out of which we received 14 complete surveys. The survey comprised of the question, the corresponding factoid answer, machine generated informative answer, and the triples used to generate the informative answer. As the generated answer includes only the triples which are lexicalized, the triple collection did not contain any triple which was not used in the lexicalization. The participants were asked to rank each generated answer based on the previously described criteria of Readability, Accuracy and Informativeness using a five point scale (Likert, 1932) as

²The survey was carried out under ethics application 16/169 with the approval of AUTEK

Q3. How many students does the Free University in Amsterdam have?

Answer: 22730

Generated Answer:

The Free University in Amsterdam has 22730 students.

Vrije Universiteit is an educational institution. It is located in Amsterdam. Its mottoes are Auxilium nostrum in nomine Domini, and Our help is in the name of the Lord. Its endowment is \$420 million.

	1	2	3	4	5
Readability and Clarity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Accuracy and Appropriateness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Informativeness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 4.15 Human ranking based evaluation survey. Each criteria was provided with a five point scale where raters can select accordingly.

shown in Fig 4.15. Each scale item was also given a numerical value and average of all rater's values was taken as the final rating.

4.3.2 Results of the Human Ranking

We first calculated the inter-rater agreement between the participants as significant disagreements can lead to major errors in the analysis. This was carried out using the Cronbach Alpha and also accompanied a scaling analysis if there are any individual participants who disagree with the majority of the raters. The result of this statistical analysis showed that there is no significant increase or decrease if a participant is removed from the collection. Therefore, ratings from all 14 participants were considered for the analysis. The inter-rater agreement analysis resulted in Cronbach Alpha values

of 0.872, 0.869 and 0.896 for readability, accuracy, and informativeness respectively. This shows a good agreement between the raters and it is therefore acceptable to make decisions based on the analysis of this data.

Figure 4.16 shows the results of the human ranking evaluation for all three criteria, readability, accuracy, and informativeness. The questions for which the answer sentence generation module was unable to generate answer sentences were not part of this evaluation as they do not form answers expected from the framework. The human ranking average values for the rest of the questions (41 questions out of 52) are depicted in the figure. It is clear from the figure that most of the questions have a low readability level while accuracy and informativeness have higher values compared to readability. This is because the readability is a criteria that is difficult to achieve as there is always an opportunity to improve the lexicalization, aggregation, and referring expressions targeted towards more readable sentences. Although we implemented these in a fashion that can generate human-like sentences, the output does not have the exact “look and feel” of the creativity that human linguistic facility provides. In addition, some missing aggregations may also have affected the readability of the answer. For instance, two sentences such as “Virje Universiteit is a university” and “It is located in Amsterdam” were not aggregated in one of the test questions that led to a less readable passage. Apart from this, redundant information mentioned in another test question was aggregated into a sentence which affected the readability of the answer. In essence, the foundation date and the foundation year for the entity “Microsoft” were aggregated by generating the sentence “The company was founded on April 04, 1975 in 1975” which is less readable and contains redundant information.

On the other hand, the accuracy is a function of content quality which focuses on how information in the triple is represented in the generated sentence. This not only

deals with the lexicalization, it also takes into consideration aggregation which deals with constructing a full sentence or sentences from a list of clauses.

Finally, informativeness has achieved higher values in many questions (30 questions with average ranking value of 4.0 or above), although it has achieved low values in some questions. For instance, Q-32 has shown the lowest informativeness score where very limited information on “Statue of Liberty” is presented and it is the shortest answer generated in the whole test question set. We also noticed that participants have rated the informativeness factor fairly independent from other factors. The best example is Q-5, which had low readability and accuracy values, but had a comparatively high informativeness score.

Figure 4.17 shows the categorized summary of human ranking based evaluation results. The results confirm the aforementioned analysis where readability of majority of the questions reside in the range 3-4, while accuracy and informativeness have majority of the questions in the range 4-5. Furthermore, there are two questions which are rated in range 2-3 for the readability criteria. As an overall evaluation, the framework performed at an acceptable level for all three criteria, although it has not achieved the highest values in all three criteria for all the questions. Specifically, it was noted that further realizations (gender and person active realizations) mostly contributed to the accuracy in a significant manner, while the number of acquired and lexicalized triples contributed to the informativeness factor. This shows the importance of post-processing tasks (e.g., realization) implemented in different modules in different granularity.

4.3.3 Post hoc Analysis

The post hoc analysis focused on identifying whether there is a possible correlation between evaluation criteria; readability, accuracy, and informativeness. Table 4.22 shows the correlation between the three evaluation criteria. The results show that there

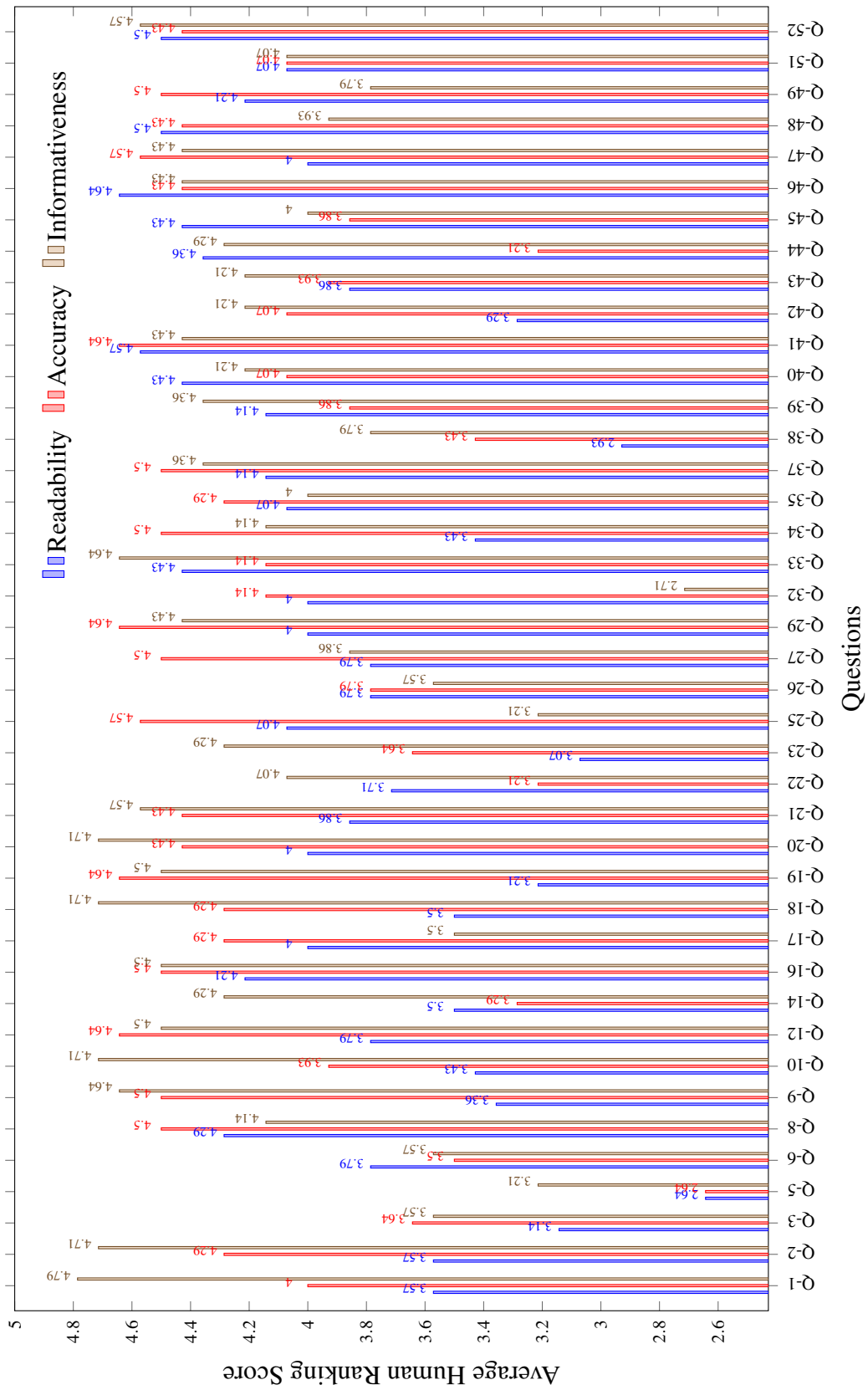


Fig. 4.16 Human ranking based evaluation results for readability, accuracy, and informativeness

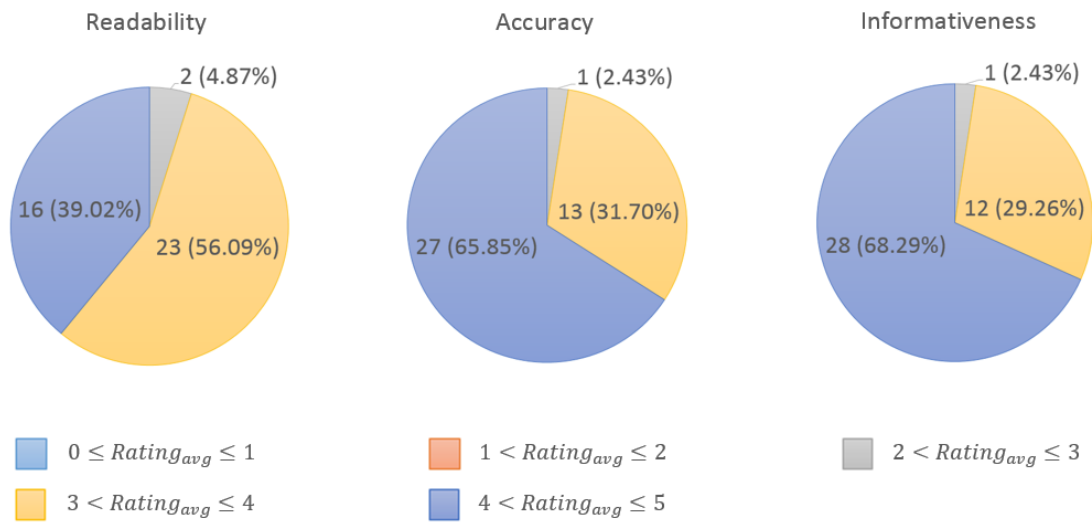


Fig. 4.17 Classification of questions into five ranges based on the human ranking evaluation. Since no record falls under the first two ranking ranges ($0 \leq Rating_{avg} \leq 1$ and $1 < Rating_{avg} \leq 2$), they are omitted in the chart.

is no significant correlation between the three evaluation criteria. This may be due to some of the reasons discussed below.

Answers which are accurate based on the content quality may still not achieve high readability level as readability depends on the suitability of aggregation, lexicalization, and referring expressions. For example, consider a scenario where an accurate sentence is aggregated with another sentence making a longer sentence which is difficult to read. Although such aggregation does not affect the content quality (accuracy), the readability can decrease significantly. This can also be the reason behind the absence of a correlation between the readability and informativeness.

Furthermore, accurate answers may still not achieve high informativeness in all scenarios as there can be specific information expected by the human participants. For instance, certain important information related to an entity may be eliminated because of the absence of a lexicalization pattern which ultimately transforms the answer to a low informative answer, although the content is highly accurate.

Table 4.22 Correlation analysis between readability, accuracy, and informativeness

	Readability	Accuracy	Informativeness
Readability	1.000	0.306	0.025
Accuracy	0.306	1.000	0.293
Informativeness	0.025	0.293	1.000

4.4 Automatic Metric based Evaluation: An Investigation

This section describes the investigation into employing automatic metrics in the answer presentation using machine generated informative answers. The next two sections describe the evaluation process, the metrics used, and their results. The feasibility of these automatic metrics is described in Section 4.4.3 using a post hoc analysis by analysing the correlation between the automatic metric results and the human rankings achieved in Section 4.3.

4.4.1 Evaluation Process

The evaluation process first calculated the similarity between the answers generated by the system and a set of answers provided by the participants for the same test question set. The results of this evaluation is analysed in Section 4.4.2. We then further analysed the suitability of these metrics as an alternative to human evaluation. Section 4.4.3 discusses the two different methods we employed to get an insight into using automatic metrics as an alternative to human evaluation; firstly we carried out a visual alignment and then we measured the correlation between the human rankings provided in Section 4.3 and the automatic metric values.

The main requirement for the automatic metric based evaluation was to compare it with the human provided answers. Therefore, we prepared a survey where the

Question: How many students does the Free University in Amsterdam have?

Answer: 22730

Answer Sentence	
Entity	Vrije Universiteit
Triples	<Vrije Universiteit, type, educational institution> <Vrije Universiteit, city, Amsterdam> <Vrije Universiteit, motto, Auxilium nostrum in nomine Domini> <Vrije Universiteit, motto, Our help is in the name of the Lord> <Vrije Universiteit, endowment, \$4.2E8 >
Description for the entity in a paragraph	

Fig. 4.18 Sample survey question to collect human answers

participants were asked to provide informative answers for the questions using the triples provided. Figure 4.18 shows an example of the survey for a sample question. The participants were also given the instruction that they can come up with the lexicalizations they prefer for the triples including a standard format for verbalizing dates, measured predicates, and other numerical values.

The human answers were only collected for the questions where the framework generated a complete answer. Therefore, the 11 questions where the framework failed to generate an answer sentence were removed from the test collection. For each of these questions, we collected two human answers resulting in 82 responses. In the first round the human answers included three invalid responses where participants have provided partial answers. Therefore, three more answers were collected in a later round to fulfil the need.

The automatic evaluation was carried out using five metrics, namely, METEOR, WAcc (1-WER), BLEU, and ROUGE-N. Chapter 2 described the details of all the aforementioned metrics. From these metrics, only BLEU and ROUGE-N are recommended to be used with paragraph size text while metrics like METEOR is recommended for single sentences. However, in this investigation, we used all the metrics and BLEU and ROUGE-N were used with different n-gram options to get the best n-gram performance.

4.4.2 Results of the Automatic Evaluation

This section describes the results of the automatic evaluation which was carried out by four different metrics using different configurations. For this evaluation we focused only on the 41 test questions which were associated with both the answer sentences and entity descriptions.

Figure 4.19 depicts the results of the automatic evaluation for the four metrics used. In this experiment we used the BLEU and ROUGE with quadgrams (latter experiments report the results of BLEU and ROUGE with four different n-gram configurations). It is evident according to Fig. 4.19 that all four metrics have given relatively similar scores. The highest scores of all four metrics were reported in Q-32 while the lowest values were reported in different questions. The lowest value for BLEU4 and ROUGE4 were reported in Q-22 and METEOR in Q-46 and WAcc in Q-34. We then analysed the standard deviation between the METEOR and WAcc values in Q-22 and their individual lowest values. For METEOR and WAcc these values were 0.021 and 0.028. These very low standard deviations show that all four metrics have performed similarly, however, this does not confirm a very strong pattern. When considering the whole question dataset, averages for all four metrics ranged from 0.21 to 0.37 with standard deviations ranging from 0.06 to 0.14. This indicates that the automatic metrics have produced very low values for the test question set.

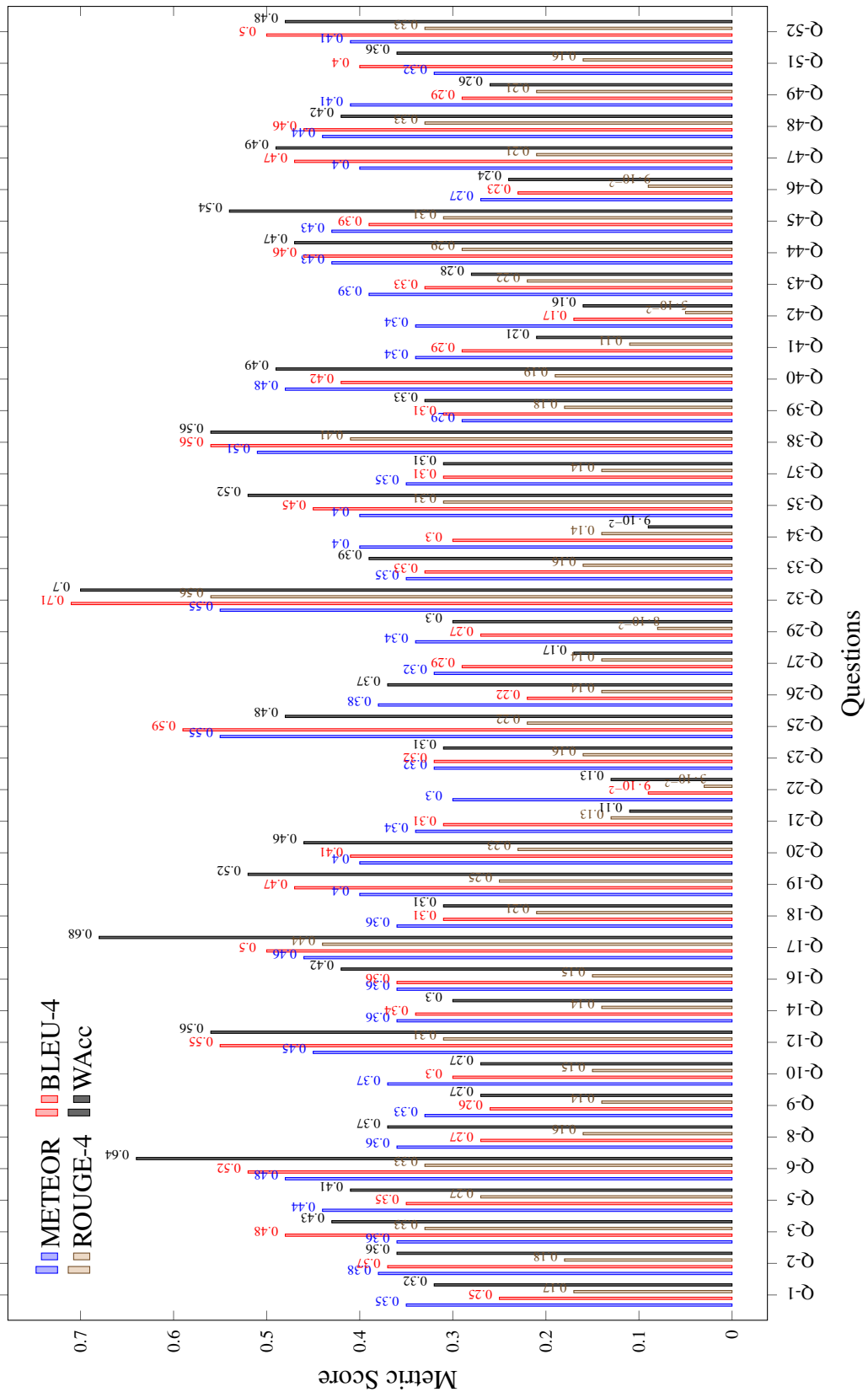


Fig. 4.19 Results of the automatic evaluation for four metrics. BLEU and ROUGE are configured to use quadgrams.

The metrics such as BLEU and ROUGE can be evaluated with different ngram settings. In this research, they were evaluated from unigram to quadgrams. Although there is no restriction to go beyond the quadgrams, such an approach will attempt to compare large text chunks which will produce erroneous results as all the answers which contain the same content but with different combinations of words will result in low values.

Figure 4.20 reports the results of the BLEU evaluation with unigrams to 4-grams. It is evident from the results that when increasing the gram size, the score has been decreased. The main reason for this is that with higher level ngrams, it has become difficult to map the human reference text to system generated text as large chunks. This is generally expected as human provided answers can have different variations in language. Since BLEU does not check for synonyms or lemmatized words, mapping exact word sequences is rather difficult to achieve. Also in an area like answer presentation there is a very high probability of providing the same semantics through different word combinations and with a high language variety.

Consecutively, we also carried out an experiment on ROUGE using unigram to 4-grams in the same way as BLEU. This also offered the same insights as the BLEU ngram experiment. When increasing the number of grams, there was also a clear decrease in the score as depicted in Fig. 4.21. The aforementioned reason also caused this behaviour in ROUGE metric with the increase in ngrams.

Since the aforementioned experiments showed that metrics perform in a similar way but with slightly different scoring levels, we carried out an inter-metric correlation test to analyse whether these metrics behave with a significant similarity. Table 4.23 reports the results from the inter-metric correlation analysis (Spearman correlations) for the main metrics with different ngram size options for BLEU and ROUGE. All the correlations were significant at the 0.01 (2-tailed) level. The lowest correlation

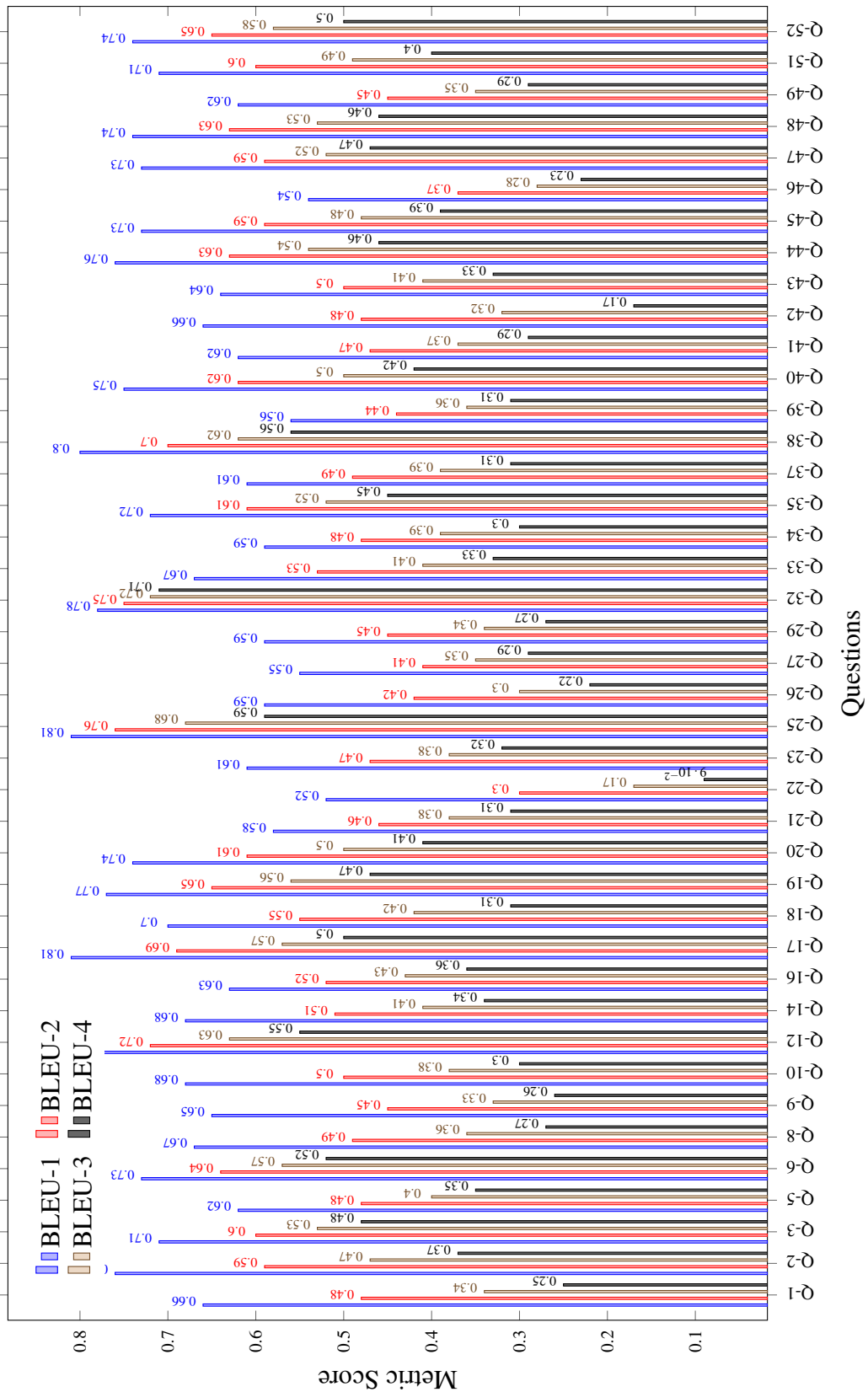


Fig. 4.20 BLEU metric results using unigrams to quadgrams

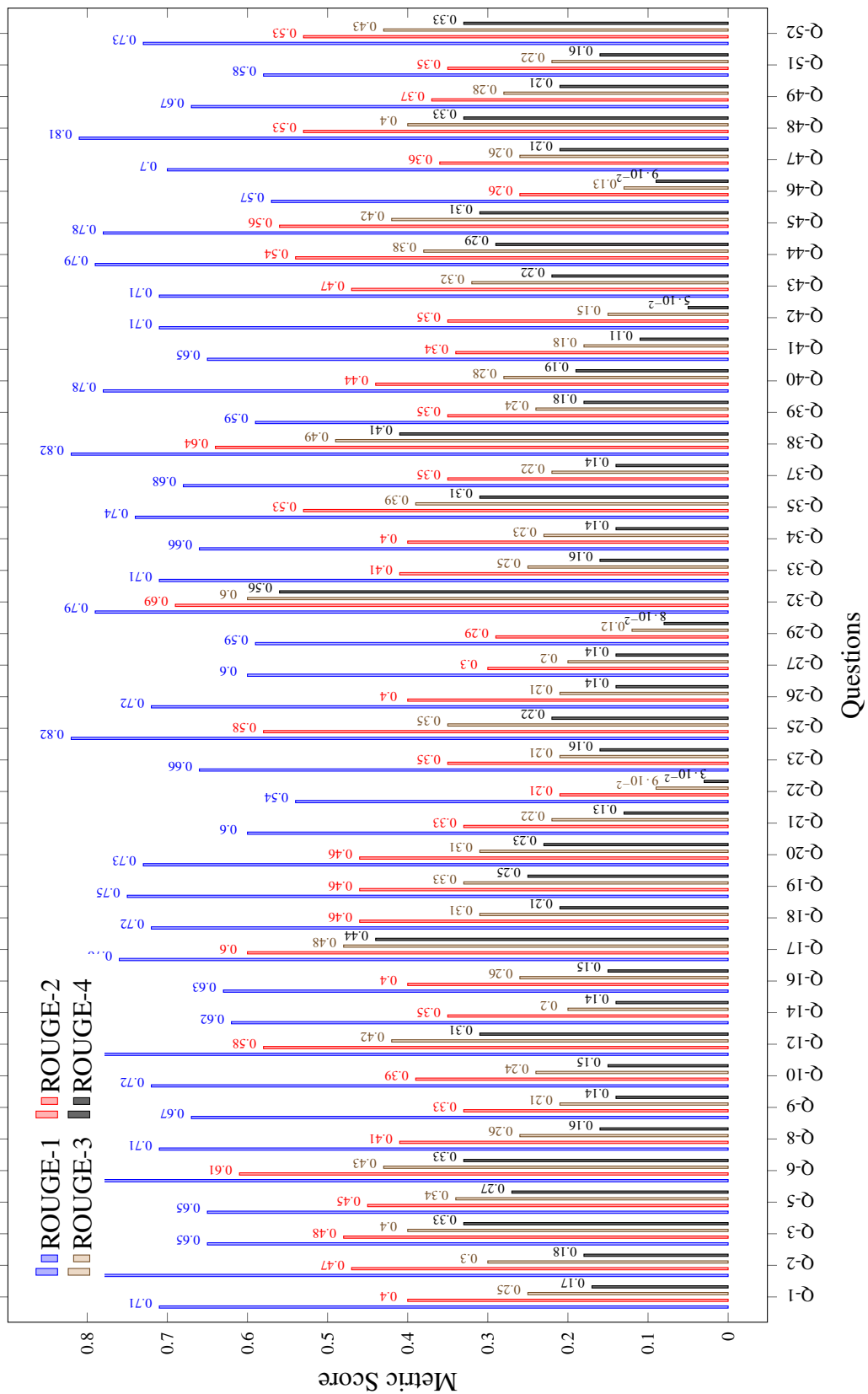


Fig. 4.21 ROUGE metric results using unigrams to quadgrams

coefficient identified was between the ROUGE-1 and BLEU-4. This may be mainly due to the ngram sizes that they consider. As ROUGE-1 is attempting to score based on the unigram level which focused on single tokens, BLEU-4 measures the quadgrams which take 4 consecutive tokens as one chunk. This can result in a reasonable difference between the scores. However, the correlation between these two metrics is acceptable although it is not strong (0.646). The highest correlation (0.984) was between the BLEU-4 and BLEU-3, where the BLEU metric is configured using two settings of trigram and quadgrams. It is also reasonable as the same metrics can perform slightly similarly when ngram sizes are very close. It is also clear when analysing correlations of the same metrics with different ngram sizes that adjacent ngram sizes show strong correlations.

4.4.3 Post hoc Analysis

The post hoc analysis focused on two areas. We first analysed the answer sentences using a visual alignment provided by METEOR to find out how human reference answer sentences deviated from the machine generated answer sentences. This was carried out only for answer sentences as it is not meaningful for large text (i.e., full answer comprised of answer sentence and entity descriptions) because METEOR alignments are focused on short sentences and do not consider large paragraph level text segments. Secondly, a feasibility analysis of automatic metrics was performed by measuring the correlation between the human rankings and the automatic metric provided values.

4.4.3.1 METEOR Visual Alignments for Answer Sentences

The METEOR visual alignments can reveal how different words/phrases in the human reference answer sentences are aligned to the system generated ones. Therefore, this can be used to understand how human answer sentences differ from the system generated

Table 4.23 Inter-metric correlation matrix for the automatic metrics. Note that * sign indicates that correlation is significant at the 0.01 level (2-tailed).

	METEOR	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE1	ROUGE2	ROUGE3	ROUGE4	WAcc
METEOR	1.000*	0.765*	0.780*	0.774*	0.749*	0.827*	0.866*	0.842*	0.787*	0.745*
BLEU1	0.765*	1.000*	0.950*	0.876*	0.835*	0.834*	0.828*	0.790*	0.771*	0.810*
BLEU2	0.780*	0.950*	1.000*	0.964*	0.923*	0.791*	0.863*	0.838*	0.808*	0.839*
BLEU3	0.774*	0.876*	0.964*	1.000*	0.984*	0.705*	0.845*	0.860*	0.843*	0.835*
BLEU4	0.749*	0.835*	0.923*	0.984*	1.000*	0.646*	0.806*	0.844*	0.849*	0.848*
ROUGE1	0.827*	0.834*	0.791*	0.705*	0.646*	1.000*	0.866*	0.771*	0.714*	0.720*
ROUGE2	0.866*	0.828*	0.863*	0.845*	0.806*	0.866*	1.000*	0.953*	0.907*	0.827*
ROUGE3	0.842*	0.790*	0.838*	0.860*	0.844*	0.771*	0.953*	1.000*	0.970*	0.834*
ROUGE4	0.787*	0.771*	0.808*	0.843*	0.849*	0.714*	0.907*	0.970*	1.000*	0.844*
WAcc	0.745*	0.810*	0.839*	0.835*	0.848*	0.720*	0.827*	0.834*	0.844*	1.000*

Table 4.24 Statistics of the alignment

Factor	Answer sentence count
Aligned with token matching (exact order)	22
Aligned with token matching (different order)	15
Aligned with stem matching	0
Aligned with synonym matching	3
Aligned with phrase matching	1

	Sys	Margaret	Thatcher	was	a	chemist
Hum						
Margaret		•				
Thatcher			•			
was				•		
a					•	
chemist						•

Fig. 4.22 Complete alignment of human reference and system answer sentence. The answer sentences are related to the question “Was Margaret Thatcher a chemist?”

answer sentences. We carried out the visual alignment process for all 41 questions that the system was able to generate answer sentences for. In the following discussion we discuss some of the significant scenarios found in the visual alignments. Table 4.24 reports the statistics of the METEOR alignment phase. It is clear according to the statistics that 53.65% were exact alignments where the system and human answer sentences had a one to one alignment.

Figure 4.22 depicts a scenario with an exact alignment between the human reference and the system generated answer sentence. This answer sentence was provided as a result for the question “Was Margaret Thatcher a chemist?”, where it is straightforward to generate an answer sentence by prioritizing the nominal subject of the question.

However, 36.58% from the 41 questions were associated with answer sentences which are not exactly aligned as above, but has the same token set in different orders. Figure 4.23a and Fig. 4.23b depict two scenarios of alignments by matching tokens in a different order. Both answer sentences have the nominal subjects of the question at the

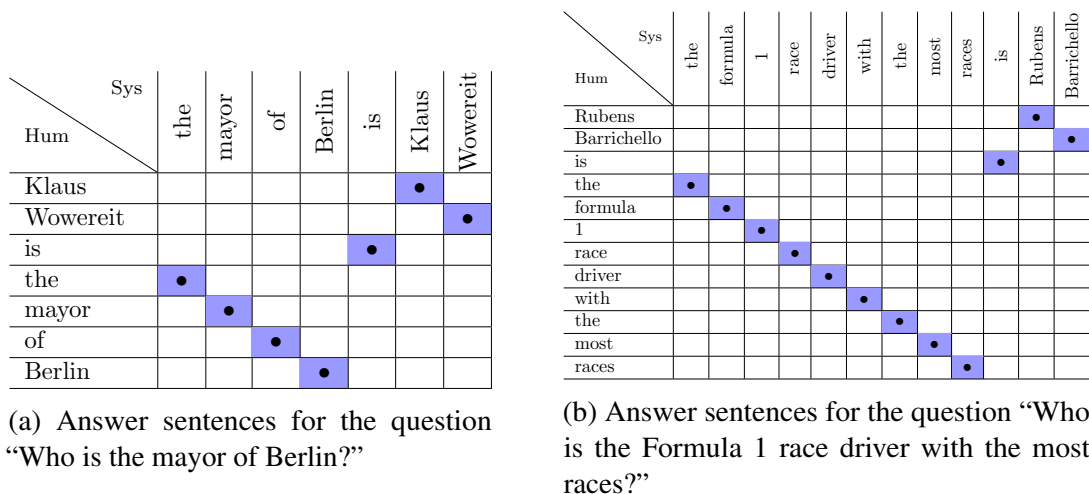


Fig. 4.23 Answer sentence alignment with different positions

start as the answer sentence generation system follows the SVO pattern and prioritizes the nominal subject. However, the human participants have provided an alternative version of the answer sentence by moving the nominal subject of the question to the end of the answer sentence.

METEOR synonym matching in the alignment phase has occurred in only three questions representing 7.31% of the answer sentences generated for 41 questions. Figure 4.24 depicts one of the synonym matching where numerical value mentioned in the system answer sentence was matched with its literal interpretation provided by one of the human participant. Although, METEOR is capable enough to match synonyms based on the lexicons provided, we only noticed three occasions where synonym matching has occurred. This is mainly because for human participants it is straightforward to provide answer sentences using the same tokens provided in the question without introducing new lexical items.

The results of the METEOR visual alignments also reported only one instance of phrase based matching. Figure 4.25 shows this instance where phrases “the husband of” and “married to” are matched using the phrase matching provided by the METEOR using the phrase lexicon. This also shows that the number of alternative ways that

		Sys				
		Benjamin	Franklin	had	3	children
Hum	Benjamin	•				
	Franklin		•			
	had			•		
	three				•	
	children					•
		Benjamin	Franklin	had	3	children

Fig. 4.24 Alignment with synonym matching. The answer sentences are related to the question “How many children did Benjamin Franklin have?”.

		Sys							
		the	husband	of	Amanda	Palmer	is	Neil	Gaiman
Hum	Amanda				•				
	Palmer					•			
	is						•		
	married	•	•	•					
	to	•	•	•					
	Neil							•	
	Gaiman								•
		the	husband	of	Amanda	Palmer	is	Neil	Gaiman

Fig. 4.25 Alignment with phrase matching. Answer sentences are related to the question “Who is the husband of Amanda Palmer?”.

answer sentences can be generated preserving the same semantics embedded on the question. Currently our framework focuses on utilizing the same linguistic structure and lexicon in the question to generate answer sentences and does not include such alternative ways of answer sentence generation using external lexicons as depicted in this example. However, such an approach is a future enhancement that will be discussed further in Chapter 5.

4.4.3.2 Feasibility Analysis of Automatic Metrics for Machine Generated Answers

This section focuses on the analysis of the feasibility of the automatic metrics by carrying out a correlation test between human rankings and the automatic metric values. The evaluation framework used here is based on the recent study by Reiter and Belz

Table 4.25 Correlation values (HR_{Corr}) between the human ratings for readability and automatic metric values.

Metric	<i>METEOR</i>	<i>BLEU1</i>	<i>BLEU2</i>	<i>BLEU3</i>	<i>BLEU4</i>	<i>ROUGE1</i>	<i>ROUGE2</i>	<i>ROUGE3</i>	<i>ROUGE4</i>	WA_{cc}
HR_{Corr}	0.049	0.034	0.080	0.079	0.061	0.050	0.023	0.057	0.033	0.138

(2009) who performed a similar feasibility study in the Machine Translation (MT) domain. From the three criteria in the human evaluation, only the readability (refers to both readability and clarity) focused on the linguistic quality of the generated answers. On the other hand the automatic metrics were also measuring the same linguistic quality using the human reference texts. Therefore, the hypothesis was that if automatic metrics measures are capable of evaluating this aspect of answer presentation, then they must show a significant correlation with averaged human rankings provided for the readability criteria.

Table 4.25 reports the correlations between the averaged human readability ratings and the scores provided by the automatic metrics. According to the reported results there were no significant correlations and all correlation coefficients were significantly low. One of the reasons for this significantly low correlation is the linguistic variety of the natural language expressions. The same semantics associated with a triple can be expressed in multiple ways using natural language and hence each informative answer to a question can vary from others due to this variety. The current automatic metrics do not consider this linguistic variety during evaluation. This is mainly because these metrics are initially designed to work in domains such as MT which often deals with very low linguistic variety as sentence to sentence translation and individual sentence level evaluation is possible. However, the informative answer generation deals with free-form natural language generation and users may express triples in different forms as well as aggregating them using multiple forms and applying referring expressions

based on their preference. Therefore, the current investigation into usage of automatic metrics in answer presentation signals to future researchers that there is a clear and urgent need to develop more suitable and reliable automatic metrics for domains which deal with free form text generation such as answer presentation.

4.5 Some Comparisons with Examples

This section focuses on comparing the RealText with existing similar answer presentation approaches with some of the examples. Since the approaches which will be mentioned here are already explained in the Chapter 2, the methodological details will not be described here unless necessary.

It is clear that RealText is capable of accessing information units instead of sentences as carried out in the approach presented by Bosma (2005). However, answers that are presented in Bosma's approach can be longer and more readable ones as they are extracted from a human provided text. The example sentence taken from Bosma's experiment is shown in Table 4.26 with highly readable text segment generated by RealText. The significant feature noticed is that sentences in Bosma's approach can condense number of information units in one sentence which is also readable as they are generated by humans. On the other hand, the RealText approach is fully automated and contains no human interaction. Therefore it has not reached the ultimate human readable form as expected from a human. However, Bosma's approach has no value towards developing fully automated QA systems as it is heavily dependent on human generated sentences.

Similarly, MedQA also has the same advantage of presenting more readable answers as it is also based on extracting sentences from human produced text. However, the same aforementioned issues also occur as this decreases automation of the QA systems. On the other hand, MedQA utilize a text based extraction of semantic information

Table 4.26 Comparison of sample answers from Bosma's (2005) approach and the RealText framework

	Bosma's (2005) approach	RealText approach
Question	What can be the cause of RSI?	Which U.S. State has the abbreviation MN?
Generated answer	A possible explanation of the development of RSI as a result of frequently repeated movements which are performed with low exertion is that the movement always involves contraction of the same muscles.	Minnesota has the abbreviation MN. Minnesota is an administrative region. It is a state in United States. Saint Paul, Minnesota is the capital of it. Its area total, area land, and area water are respectively 225.2 billion m ² , 206.375 billion m ² , and 19 billion m ² . Its minimum elevation, and maximum elevation are respectively 183.0 m, and 701.0 m.

and RealText uses the structured data based semantic information. The advantage that RealText achieves using this structured form is that ambiguity is already resolved compared to text, where ambiguity represents a serious hurdle when linking semantic information. For example, the question "Does the new Battlestar Galactica series have more episodes than the old one?" refers to two entities that are very similar in content but points to two different entities, namely, "Battlestar Galactica (1978 TV series)" and "Battlestar Galactica (2004 TV series)". An approach like MedQA will face certain issues in ambiguity resolution when generating answers from the text as both entities are very similar. However, this is overcome in RealText by following a bottom up approach for answer generation using information related to both entities and transforming them to natural language while aggregating and generating referring expressions as shown in Table 4.27.

AQUA (Vargas-Vera and Motta, 2004) used a similar approach in using structured data which are extracted from a domain ontology compared to what RealText carried

Table 4.27 RealText generated entity descriptions for a question which contains two similar entities

Question	Does the new Battlestar Galactica series have more episodes than the old one?
Generated Entity Descriptions	<p>Battlestar Galactica (1978 TV series) is a television show. It was created by Glen A. Larson. Stu Phillips composed the music for it. Jonathan Harris, Tony Swartz, David Greenan, among others, starred in Battlestar Galactica (1978 TV series). Its runtime is 45.0 minutes. Number of episodes, and number of seasons in it are respectively 24, and 1. It is aired on American Broadcasting Company.</p> <p>Battlestar Galactica (2004 TV series) is a television show. It was created by Glen A. Larson. Ronald D. Moore be executive producer of the it. Bear McCreary composed the music for Battlestar Galactica (2004 TV series). It was produced by Ronald D. Moore, and David Eick. Aaron Douglas, Grace Park, Michael Hogan, among others, starred in it. Its runtime is 44.0 minutes. Its format is 1080i. Number of episodes, and number of seasons in it are respectively 75, and 4. It is aired on Syfy.</p>

out in an open domain large scale Linked Data cloud. The disadvantages of using a domain ontology was already discussed in Chapter 2. Table 4.28 reports two sample answers and data sources, where one taken from AQUA and the other from RealText. AQUA relies on a domain ontology and uses the same properties mentioned in the ontology to generate the answer using a controlled natural language format. On the other hand, RealText uses four different methodologies to produce lexicalizations for the linked data and also contains the aggregation and referring expression generation to improve the generated text. This has caused the RealText framework to generate text content with high language variety compared to AQUA.

Intentional answer generation system, WEBCOOP (Benamara, 2004), also focuses on the answer presentation based on lexicalization and using a knowledge base. The lexicalization introduced by WEBCOOP is based on the phrase and token level lexical selection instead of a complete lexicalization of the structured data using a multi-

Table 4.28 A comparison between the answers provided by AQUA (Vargas-Vera and Motta, 2004) and RealText

	AQUA (Vargas-Vera and Motta, 2004) example	RealText example
Question	Who works in AKT?	How many employees does Google have?
Generated answer	<p>AKT is a project at KMi and each person of the AKT team is a researcher at KMi.</p>	<p>Google has 49829 employees. Google is a company. The company was founded on September 04, 1998 in Menlo Park. It is located in Mountain View. Its equity is \$87300 million. Its net income, operating income, and revenue are respectively \$12.9 billion, \$13.96 billion, and \$59820 million. It released list of Google products. It employs 49829 employees.</p>
Data source		

strategy approach as in RealText. On the other hand, RealText works towards the aim of generating informative answers and therefore extracts all the related information of entities mentioned and transform them to natural language. WEBCOOP is not intended towards an informative answer generation, however, focuses on providing further solutions as in links.

The intentional answer generation system explained by Cimiano et al. (2008) is focused on providing additional information for a question in logical form. However, this logical answer is not realized to a natural language answer and it is mentioned by Cimiano et al. (2008) as a future goal of the research. Table 4.29 shows an example of answers generated by Cimiano et al.'s approach and RealText in similar themes. Cimiano et al.'s approach for answer presentation focuses on the intensional aspects by generating generalizable answer in logical form. In comparison, RealText generates an informative answer with information related to the entities. However, given a question like "which states have a capital?", RealText will be able to generate entity descriptions for all the answers (i.e., states) which is not useful as this is a list based question. This is the main motivation that RealText is only focusing on factoid questions and do not consider list based or definitional questions.

4.6 Limitations and Assumptions

Despite the novelty that RealText brings to the answer presentation domain, it has limitations in scope and it is based on some pre-existing conditions. Firstly, RealText does not take into consideration the consistency and accuracy of the Linked Data. The accuracy and consistency of the data in DBpedia has been investigated by a number of researchers, and solutions such as the DBpedia mapping project (Lehmann et al., 2014) and crowdsourced quality evaluation approaches (Kontokostas et al., 2013) have been introduced to improve the accuracy and the consistency of the data. For instance,

Table 4.29 A comparison between the answers provided by Cimiano et al.'s (2008) approach and RealText

	Cimiano et al.'s (2008) approach	RealText
Question	Which states have a capital?	What is the capital of Canada?
Generated answer	answer (X) ←state (X)	The capital of Canada is Ottawa. Canada is a country. Ottawa is the capital city of it. Largest city in it is Toronto. Canadian dollar is ... Ottawa is a city. It is the capital of the Canada. It is part of National Capital Region , and Ontario. Ottawa is founded on ...
Data source	Knowledge base/ Linked Data	Linked Data

the DBpedia mapping project introduced the consistent terminology for the predicate naming and resolved inconsistent naming of the predicates under the DBpedia property schema which is now deprecated. Extensive quality evaluation by Färber et al. (2016), which covered a number of areas (accuracy, consistency, completeness, timeliness and several other factors) also confirmed that DBpedia is a good quality data source. The data accuracy may affect accuracy and readability of the generated answer as well as the informativeness as users may judge inaccurate information as uninformative. In addition, Table 4.18 reported some of the content issues in DBpedia that lead our lexicalization process to generate incorrect interpretations of the semantics.

The answer sentence generation module of the framework focused on generating an answer sentence based on the dependency tree and following the SVO form of English. However, there are alternative ways to generate the answer sentence with different order in tokens used in the source question. This is also confirmed by the human provided answer sentence where we noticed readable and accurate answer sentences which still utilize the source question tokens and structure. This language variety in

answer sentence generation is not currently handled by the framework and remains as a future goal.

Relational patterns in the lexicalization module aligned the triple with the text that is extracted from Wikipedia or WWW. However, in some scenarios, the actual entity was not mentioned throughout the text. For example, the entity, Secret Intelligence Service, mentioned in the question set had an abbreviation called “SIS” as well as an acronym called MI5. This has caused the co-reference resolution process to map the entity name to each sentence that describes the entity and consequently failed during the relation-triple alignment as the triple subject is not mentioned in the relation in most cases. This is mainly because of the limitation of the research that it currently does not analyse the text and associate the acronyms and abbreviations of the entity.

The only three referring expressions used were personal pronouns, ontology classes, and a part of the names. There is also an opportunity to enhance the generated answer by associating acronyms (e.g., MI5 = Secret Intelligence Service) and abbreviations (e.g., MIT = Massachusetts Institute of Technology). However, currently the Linked Data cloud does not contain such extensive knowledge on different aliases that can be used and for some entities these aliases are significantly long as they are full names of the entity being described.

Another limitation of the research is that the realization module transforms sentences to past tense only for people who are not alive. This cannot be accomplished for other entities due to a lack of predicates (i.e., a predicate like “death date” in *Person* ontology class) to identify such entities which are currently existing or not. For instance, consider a DBpedia entity of an organization which is not currently operating. There is no predicate to identify whether the organization is closed or when the organization is closed. Therefore, it is not possible to transform the sentences related to these entities to past tense. Furthermore, the realizer works only in one direction and does not consider

transforming a past tense sentence to present if necessary. This is also due to a lack of predicates to identify whether such position or state is still being held by the entity (e.g., a management position held by a person). For example, consider the technology advisor position held by Bill Gates in Microsoft. Currently DBpedia does not record the time period of which Bill Gates held the technology advisor position at Microsoft. Therefore, it is difficult to transform to the correct tense without this temporal data about the triple.

4.7 Chapter Summary

This chapter described the evaluation of the RealText framework which focused on generating informative answers for QA systems utilizing NLG and Linked Data. The evaluation of the framework was described in two aspects; firstly we discussed the module wise evaluation covering all the main modules in the RealText framework and secondly a human evaluation was carried out to investigate the readability, accuracy and the informativeness of the answers. In addition, we also carried out an automatic metric based evaluation where the feasibility of four metrics were evaluated with different settings. The human evaluation which was used to determine the overall performance of the framework confirmed that RealText can generate highly readable, accurate and informative answers. The investigative study of automatic metrics in answer presentation revealed that current automatic metrics used in other areas cannot reliably generate a score which correlates with human scores. This creates a need for future work in QA that should focus on designing automatic metrics that can accurately measure the quality of the informative answers.

The future work of this research including the one that is discussed in the previous paragraph are discussed in detail in the next chapter which concludes this thesis. In addition, the next chapter summarizes the contributions of the research and presents the concluding remarks.

Chapter 5

Conclusion

This chapter discusses the contributions, future directions, and conclusions from this research.

5.1 The Contributions of this Research

The contributions from this research can be classed into two different research areas, namely, NLP (QA and NLG) and the Semantic Web. The following discussion describes the respective contributions as well as some applications which are based on the research presented in this thesis.

For the NLP discipline, this research directly contributes to the enhancement of QA systems in terms of a framework for presentation of the answer with a “human feel”. Instead of the previous factoid presentation of the answer to a question, the framework is able to present the answer embedded in a full sentence appropriate to the question, as well as present additional sentences containing extraneous contextual information formulated as a paragraph exhibiting the tone and tenor of a human constructed paragraph. The extraneous contextual information is extracted from DBpedia triples using the entities from both the answer and the question. The triples then go through

further linguistic processes, namely, lexicalization, aggregation, referring expression generation, and additional linguistic realizations to formulate them into text which closely resembles human generated text.

The framework presented for the case study of QA systems will also eventually support a number of related areas as well. For instance, with the advent of humanoid robotics, researchers are in the process of searching for techniques to make robots which are more human-like. The research presented in this thesis directly contributes to humanoid robotics research by presenting a transferable framework which will be able to answer questions in a more natural form as well as in a manner in which human end users will be able to get contextual knowledge from the robot thus enhancing the value and realism of machines as humans. As another application, the framework will also be able to help Intelligent Personal Assistants (IPA) to be able to interact with humans with a richer interface exhibiting human properties.

In terms of the contribution to NLG systems, the research presented a detailed lexicalization module, which in essence, focused on structured data in the form of triples and a technique for using the associated underlying semantics. The realization strategies proposed could be used in a number of scenarios. For example, it could be used to check the syntactic accuracy of human produced text and auto-correct them for gender mismatch (gender realization) or for the correct tense for a person who is not alive (active person realization). Similarly, the aggregation techniques could also be used to identify if the text needs to be presented as an integrated single sentence akin to a human generated sentence. The referring expression generation module uses world knowledge to enhance the generation of referring expressions. DBpedia ontology was used as a shared conceptualization of world knowledge to determine appropriate referring expressions which has an added effect of embedding additional world knowledge in the generated text.

The Semantic Web is generally thought of as a method for transforming the document web into a data web using structured data, enriched with embedded semantics. In this research, we introduced the idea of using the Semantic Web in reverse, that is, using it to generate natural language based documents. The technique presented a generic approach for generating text, not limited to just answer presentation. The overall effect is that it bridges the gap between humans and machines by transforming machine friendly Semantic Web data into a version of human friendly natural language text, which is easily understood by humans. This solution can be used in a number of scenarios, some of which are briefly discussed below.

The Semantic Web to natural language transformation approach could also be applied to improve human-computer interaction in information kiosks. For instance, in a museum a user may want to know information related to an item and other items linked to the same time period. The information in these kiosks is output in a natural language form for which the information is typically extracted from free texts hence is prone to errors, such as ambiguity. Encoding the information in a linked structured data form substantially reduces the manifestation and perpetuation of such errors. A framework such as the one proposed can then be used to transform the triples into natural text so that it does not lose the “human feel”. The contribution of this research in this area is to generate a textual representation of this data by constructing texts which emanates the tone and tenor of a human generated language. Similarly, the presented framework could also be utilized in areas such as eLearning where students could be provided with a textual representation of information which is concise and accurate, rather than a set of web pages as is currently the case with the output from search engines. In effect, the framework could be extended to numerous other fields such as journalism, eHealth, military information management, inter alia, which need machine interaction with a knowledgebase.

5.2 Future Works

The research currently employs DBpedia as the main Linked Data resource which acts as the source of information. However, a possible future extension of this research would be to investigate the use of other Linked Data resources together with DBpedia as an ensemble source of contextual knowledge in order to widen the knowledgebase. Specifically, Linked Data resources from closed domains could be used to generate informative answers in closed domain QA systems, such as Biomedical systems. This would also require a number of additional language resources in addition to those that have been used in this research, which focused on open domain QA.

The answer sentence generation module of the framework employs dependency subtree patterns to generate the answer sentence. In future, the answer sentence generation could also be extended to focus on generating sentences with languages other than English. Specifically, one of the prioritized future tasks is to find ways in which the same linguistic structure could be transformed to an answer sentence in another language. This will enable us to investigate different forms of the same dependency parsed question which could be transformed into a sentence with an embedded answer. This will also play a significant role in understanding another language. In addition to the different parsed forms, another interesting aspect would be to investigate the use of similar phrases and synonyms to generate answer sentences so that exactly the same sentence would not be generated twice. This would give even more of a human feel to the generated answer. To do this, one would require a lexicon of phrases and synonyms (e.g., from WordNet) and an algorithm to substitute the tokens with appropriate ones while still preserving the semantics and pragmatics associated with the original question. Furthermore, one of the key tasks would be to apply answer sentence generation to languages other than English, to see the viability of the method. Initially, this could be carried out in languages which closely resemble the linguistic structure in English, such

as Romance languages including French and Spanish which follow the SVO structure (Harris and Vincent, 2003).

In terms of lexicalization, further research could explore additional pattern mining processes that would contribute to the current ensemble of pattern extraction processes. Additionally, the existing pattern extraction modules could be refined in order to increase the accuracy of the generated patterns. The relational pattern module is one of the principal modules that is prone to errors as it extracts patterns from the unstructured text. It is challenging to improve these, however, a future study could investigate further refinements to the pattern extraction process and post-hoc realizations in order to reduce some of the errors. The improvements in relational patterns could also focus on utilizing better alignment between the relations and triples. Specifically, this will require the use of acronyms and abbreviations associated with subjects as it is one of the missing features in the current framework. Furthermore, the pattern realization process could also be improved in the future as it has already been proven in this research that it could resolve a number of errors. In this research, we only focused on the active realization for a person who is not alive. This can be enhanced in the reverse direction in order to realize a past tense pattern into the active form for a person who is alive. This was left out of scope in this research due to the ambiguity of the patterns for a person who is alive which may also be denoted in past tense for an event being described which occurred in the past. If we associate additional metadata with the triple to denote the time period of the activity or the event related to the triple, then this information could be used to decide whether the pattern should be realized or not. Such an approach may benefit the Semantic Web as well as a number of other applications based on Linked Data. Even though this will require additional effort in improving the underlying metadata, it will add value to the overall framework.

The current aggregation module is based on a number of rules which require certain conditions to be fulfilled in order to merge multiple lexicalized triples into a single sentence. The coverage of these aggregations still needs improvement in order to identify and carry out all possible aggregations. A future study could explore other possible aggregations as well as some complex ones based on deeper semantic and pragmatic analysis. In addition, it is also important to investigate chaining current one-off aggregations to design aggregation chaining where a single sentence can be produced by aggregating multiple triples using various aggregation rules.

The current referring expressions generated are either personal pronouns or world knowledge based anaphora using the core ontology class of the entity. However, the latter is currently limited to a number of specified ontology classes. This is mainly because all of the ontology classes are not commonly used to denote an entity as a referring expression. The future studies could investigate transforming this into an automatic process of identifying whether the core ontology class is suitable to denote the given entity as a referring expression. It is also important to use the existing texts to develop a system that can learn the suitability of a referring expression automatically.

Section 4.4 reported a feasibility study of the automatic evaluation metrics and results showed that human evaluation does not correlate with any of the automatic metrics investigated. There are a number of reasons behind this which include language variety, different aggregations, and verbalizations. Since automatic metrics use human reference answers, these reference answers and system generated answers were using different phrases and a different order of referring expressions, which ultimately increased the language variety and decreased the value of the ngram based evaluation approach. Furthermore, different variations can also occur in aggregations based on the preference of the user on which properties should appear in a single sentence. For example, a number can be represented as is, in millions, or billions depending on the

user. Automatic evaluation is necessary when developing frequently changing programs which require a number of evaluations. Future work could also investigate developing metrics that can closely correlate with human evaluation by analysing all three aspects; syntax, semantics and pragmatics. Another key goal in developing such a metric is to introduce a variable that could be adapted based on the length of the text regardless of whether it is a single sentence or a paragraph with multiple sentences. This will address the drawbacks of testing with different ngram sizes in the current metrics such as BLEU and ROUGE.

There are also some extensions that can be done to the overall framework. Firstly, it is imperative to introduce a content selection method on top of the NLG layer. This content selection module should be able to select a subset of most relevant triples from the triple collection to generate text content for a user selected text length. This will serve the users by helping them to grasp the most important knowledge based on their need and time availability. The framework could also be integrated into a number of application areas and human satisfaction scores and feedback could be collected in order to identify the potential areas to improve. We have already started applying this framework into eHealth with other collaborators and in the future we expect to develop a fully-fledged eHealth application that would be able to generate text based on Linked Data triples available in the health and medical domain.

5.3 Concluding Remarks

Answer presentation is a subtask in QA which is crucial for developing human-like AI systems. Recently, with the trend towards QALD, QA systems have access to massive amounts of knowledge encoded as Linked Data. The focus of this research was a framework which can be used in QALD to present *Informative Answers* as natural language. We utilized Linked Data to acquire additional information related to the

entities mentioned in the question and the answer, and transformed them into natural language using NLG. Furthermore, the factoid answer was also presented as a full, natural language sentence based on the linguistic structure of the source question. The *Informative Answer* generated closely resembles an answer that can be expected from a human expert.

The framework presented in the thesis is composed of multiple modules which are highly cohesive, loosely coupled, and communicate through attribute value matrices. This will enable future researchers to de-assemble the framework and utilize the modules to carry out further research and apply them in a variety of applications. We have conducted a series of evaluations to investigate whether the framework satisfies the main objective of this research. The human evaluation which is the cornerstone of all NLP evaluations showed that the proposed framework can generate readable, accurate, and informative answers by achieving an acceptable human score. The individual modules were also examined separately to analyse their contribution to the main objective. The lexicalization module acts as the main module of the framework which generates the text from Linked Data, which was evaluated in depth by investigating the lexical, semantic, and overall accuracy of the lexicalized triples. All of these evaluations unequivocally showed that the framework is capable of generating readable, accurate and informative answers, which add value to answer presentation.

References

- Abele, A., McCrae, J. P., Buitelaar, P., Jentsch, A., and Cyganiak, R. (2017). The Linking Open Data Cloud Diagram. Technical report, Insight Centre for Data Analytics.
- Achananuparp, P., Hu, X., and Shen, X. (2008). The Evaluation of Sentence Similarity Measures. In *10th International Conference on Data Warehousing and Knowledge Discovery*, pages 305–316. Springer-Verlag.
- Adams, V. (2014). *Complex Words in English*. Routledge.
- Alexiadou, A. and Schäfer, F. (2008). Instrumental -er Nominals Revisited. In *27th West Coast Conference on Formal Linguistics*, pages 10–19, California, USA. University of California.
- Alexiadou, A. and Schäfer, F. (2010). On the Syntax of Episodic vs. Dispositional -er Nominals. In *The Syntax of Nominalizations across Languages and Frameworks*, pages 9–38. Walter de Gruyter.
- Allemang, D. and Hendler, J. (2008). *Semantic Web for the Working Ontologist*. Morgan Kaufmann Publishers Inc.
- Allen, C., Altaf, F., Clay, S., and Yan, S. (2016). Techniques for Answering User Questions Based on User Experience. Technical report, International Business Machines Corporation.
- Auer, S., Bizer, C., Kobilarov, G., and Lehmann, J. (2007). DBpedia: A Nucleus for a Web of Open Data. In *6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 722–735, Busan, Korea. Springer-Verlag.
- Banerjee, S. and Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics.
- Benamara, F. (2002). A Semantic Representation Formalism for Cooperative Question Answering Systems. In *International Conference on Knowledge Based Computer Systems (KBCS)*.
- Benamara, F. (2004). Generating Intensional Answers in Intelligent Question Answering Systems. In *3rd International Conference on Natural Language Generation (INLG)*, pages 11–20. Springer-Verlag.

- Benamara, F. and Saint-Dizier, P. (2004). *Advanced Relaxation for Cooperative Question Answering*. MIT Press.
- Bizer, C. (2009). The Emerging Web of Linked Data. *IEEE Intelligent Systems*, 24(5):87–92.
- Bizer, C., Lehmann, J., and Kobilarov, G. (2009). DBpedia-A Crystallization Point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3).
- Boley, H. and Possner, S. (1995). Least General Generalization. Technical report, University of Kaiserslautern, Kaiserslautern, Germany.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *ACM SIGMOD International Conference on Management of Data*, page 1247, New York, New York, USA. ACM Press.
- Bosma, W. (2005). Extending Answers using Discourse Structure. In *Recent Advances in Natural Language Processing*, Borovets, Bulgaria. Association for Computational Linguistics.
- Bundy, A. and Wallen, L. (1984). Context-Free Grammar. In *Catalogue of Artificial Intelligence Tools*, pages 22–23.
- Busemann, S. (2005). Ten Years After : An Update on TG/2 (and Friends). In *10th European Workshop on Natural Language Generation*, Aberdeen, Scotland. University of Helsinki.
- Cao, Y., Liu, F., Simpson, P., Antieau, L., Bennett, A., Cimino, J. J., Ely, J., and Yu, H. (2011). AskHERMES: An Online Question Answering System for Complex Clinical Questions. *Journal of Biomedical Informatics*, 44(2):277–288.
- Carlson, L., Marcu, D., and Okurowski, M. E. (2003). Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory. In *Current and New Directions in Discourse and Dialogue*, pages 85–112. Springer-Verlag.
- Cimiano, P., Hartfiel, H., and Rudolph, S. (2008). Intensional Question Answering using ILP: What does an answer mean? In *13th International Conference on Applications of Natural Language to Information Systems*, pages 151–162. Springer-Verlag.
- Cimiano, P., Rudolph, S., and Hartfiel, H. (2010). Computing Intensional Answers to Questions - An Inductive Logic Programming Approach. *Data & Knowledge Engineering*, 69(3):261–278.
- Clark, H. H. and Brennan, S. E. (1991). Grounding in Communication. In *Perspectives on Socially Shared Cognition*, pages 259–292. American Psychological Association.
- Clark, P. (2010). Query Relaxation in AURA. Technical report, Allen Institute for Artificial Intelligence, Seattle, USA.

- Coleman, M. and Liao, T. L. (1975). A Computer Readability Formula Designed for Machine Scoring. *Journal of Applied Psychology*, 60(2):283–284.
- Conway, D. (1998). An Algorithmic Approach to English Pluralization The problem of English plurals Categories of English plurals. In *2nd Annual Perl Conference*, San Jose, CA, USA. O'Reilly Media Inc.
- Cormen, T. R., Leiserson, C. E., and Rivest, R. L. (1989). *Introduction to Algorithms*. MIT Press.
- de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal Stanford Dependencies: A Cross-linguistic Typology. In *9th International Conference on Language Resources and Evaluation (LREC)*, pages 4585–4592. European Language Resources Association.
- Del Corro, L. and Gemulla, R. (2013). ClausIE: Clause-based Open Information Extraction. In *22nd international conference on World Wide Web*, pages 355–366. International World Wide Web Conferences Steering Committee.
- Demner-Fushman, D. and Lin, J. (2005). Knowledge Extraction for Clinical Question Answering: Preliminary Results. In *AAAI-05 Workshop on Question Answering in Restricted Domains*. American Association for Artificial Intelligence.
- Demner-Fushman, D. and Lin, J. (2006). Answer Extraction, Semantic Clustering, and Extractive Summarization for Clinical Question Answering. In *21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 841–848, Morristown, NJ, USA. Association for Computational Linguistics.
- Denkowski, M. and Lavie, A. (2011). Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *6th Workshop on Statistical Machine Translation*, pages 85–91. Association for Computational Linguistics.
- Duboue, P. A. and McKeown, K. R. (2003). Statistical Acquisition of Content Selection Rules for Natural Language Generation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 121–128, Morristown, NJ, USA. Association for Computational Linguistics.
- Duma, D. and Klein, E. (2013). Generating Natural Language from Linked Data: Unsupervised Template Extraction. In *10th International Conference on Computational Semantics (IWCS)*, Potsdam. Association for Computational Linguistics.
- Ell, B. and Harth, A. (2014). A Language-independent Method for the Extraction of RDF Verbalization Templates. In *8th International Natural Language Generation Conference*, Philadelphia. Association for Computational Linguistics.
- Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., and Vrandečić, D. (2014). Introducing Wikidata to the Linked Data Web. In *International Semantic Web Conference*, pages 50–65. Springer-Verlag.
- Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open Information Extraction from the Web. *Communications of the ACM*, 51(12):68–74.

- Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying Relations for Open Information Extraction. In *Empirical methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.
- Färber, M., Ell, B., Menne, C., Rettinger, A., and Bartscherer, F. (2016). Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal*, 9(1):1–5.
- Field, A. (2010). *Discovering Statistics Using SPSS*. SAGE Publications Inc., second edition.
- Fleiss, J. L. (1971). Measuring Nominal Scale Agreement among many Raters. *Psychological Bulletin*, 76(5):378–382.
- Fry, E. (1980). Graph for Estimating Readability–Extended. *Canadian Library Journal*, 37(4):249.
- Gaasterland, T. (1997). Cooperative Answering through Controlled Query Relaxation. *IEEE Expert*, 12(5).
- Gaasterland, T. and Lobo, J. (1994). Qualified Answers that Reflect User Needs and Preferences. In *20th Conference on Very Large Databases*, Santiago, Chile. Morgan Kaufmann Publishers Inc.
- Gatt, A. and Reiter, E. (2009). SimpleNLG: A Realisation Engine for Practical Applications. In *12th European Workshop on Natural Language Generation*, pages 90–93, Athens, Greece. Association for Computational Linguistics.
- Ginzburg, J. and Sag, I. A. (2000). *Interrogative Investigations*. Stanford CSLI Publications.
- Grammarly (2016). Systems and Methods for Advanced Grammar Checking. Technical report, Grammarly Inc.
- Gunning, R. (1968). The Fog Index After Twenty Years. *Journal of Business Communication*, 6(2):3–13.
- Harris, M. and Vincent, N. (2003). *The Romance Languages*. Routledge.
- Hatzivassiloglou, V., Gravano, L., and Maganti, A. (2000). An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering. In *23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece. ACM Press.
- Heycock, C. (2014). Generative syntax 6.1 - wh- interrogatives. Technical report, The University of Edinburgh.
- Hitzler, P., Krotzsch, M., and Rudolph, S. (2009). *Foundations of Semantic Web Technologies*. CRC Press.
- Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, USA, first edition.

- Katz, B., Borchardt, G., Felshin, S., Shen, Y., and Zaccak, G. (2007). Answering English Questions using Foreign-language, Semi-structured Sources. In *International Conference on Semantic Computing (ICSC)*, pages 439–445. IEEE.
- Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A. J., and Temelkuran, B. (2002). Omnibase: Uniform Access to Heterogeneous Data for Question Answering. In *International Conference on Application of Natural Language to Information Systems*, pages 230–234. Springer.
- Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new Readability Formulas (automated readability index, fog count and flesch reading ease formula) for Navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.
- Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2008). A Large-scale Classification of English Verbs. *Language Resources and Evaluation*, 42(1):21–40.
- Kobilarov, G., Bizer, C., Auer, S., and Lehmann, J. (2009). DBpedia - A Linked Data Hub and Data Source for Web and Enterprise Applications. In *International World Wide Web Conference*, pages 1–3. ACM Press.
- Kohlschütter, C., Fankhauser, P., and Nejdil, W. (2010). Boilerplate Detection using Shallow Text Features. In *ACM International Conference on Web Search and Data Mining*, pages 441–450. ACM Press.
- Kolomiyets, O. and Moens, M.-F. (2011). A Survey on Question Answering Technology from an Information Retrieval Perspective. *Information Sciences*, 181(24):5412–5434.
- Kontokostas, D., Zaveri, A., Auer, S., and Lehmann, J. (2013). Triplecheckmate: A Tool for Crowdsourcing the Quality Assessment of Linked Data. In *Communications in Computer and Information Science*, pages 265–272. Springer-Verlag.
- Krippendorff, K. (2004). Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research*, 30(3):411–433.
- Krippendorff, K. (2007). Computing Krippendorff’s alpha reliability. Technical report, Annenberg School for Communication, University of Pennsylvania, Philadelphia, USA.
- Kubler, S., McDonald, R., and Nivre, J. (2009). *Dependency Parsing*. Morgan & Claypool Publishers.
- Lakoff, G. and Johnson, M. (2003). *Metaphors We Live By*. University Of Chicago Press.
- Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D. (2011). Stanford’s Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Conference on Natural Language Learning*, Portland. Association for Computational Linguistics.

- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., and Bizer, C. (2014). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web journal*, 5(1):1–29.
- Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet physics doklady*, 10(8):707–710.
- Li, X. and Roth, D. (2002). Learning Question Classifiers. In *19th International Conference on Computational linguistics (COLING)*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.
- Likert, R. (1932). A Technique for the Measurement of Attitudes. Technical report, Archives of Psychology.
- Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Workshop on Text Summarization Branches Out*, Barcelona, Spain. Association for Computational Linguistics.
- Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., and Karger, D. R. (2003). The Role of Context in Question Answering Systems. In *Conference on Human Factors in Computing Systems*, page 1006, New York, New York, USA. ACM Press.
- Liu, Y. and Agichtein, E. (2008). You’ve got answers: Towards Personalized Models for Predicting Success in Community Question Answering. In *46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 97–100. Association for Computational Linguistics.
- Mann, W. C. and Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281.
- Manning, C., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore. Association for Computational Linguistics.
- Matuszek, C., Cabral, J., Witbrock, M., and Deoliveira, J. (2006). An Introduction to the Syntax and Content of Cyc. In *AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49.
- Mausam, Schmitz, M., Bart, R., Soderland, S., and Etzioni, O. (2012). Open Language Learning for Information Extraction. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island. Association for Computational Linguistics.
- Maybury, M. (2008). New Directions In Question Answering. In Strzalkowski, T. and Harabagiu, S. M., editors, *Advances in Open Domain Question Answering*, volume 32 of *Text, Speech and Language Technology*. Springer Netherlands, Dordrecht.
- McBride, B. (2002). Jena: A Semantic Web Toolkit. *IEEE Internet Computing*, 6(6):55–58.

- McLaughlin, G. H. (1969). SMOG Grading: A New Readability Formula. *Journal of Reading*, 12(8):639–646.
- Melo, D., Rodrigues, I. P., and Nogueira, V. B. (2013). A Review on Cooperative Question-Answering Systems. techreport, University of Évora.
- Mendes, A. C. and Coheur, L. (2013). When the Answer comes into Question in Question-answering: Survey and Open Issues. *Natural Language Engineering*, 19(1):1–32.
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.
- Moriceau, V. (2006). Numerical Data Integration for Cooperative Question-answering. In *European Chapter of the Association for Computational Linguistics Workshop On KRAQ Knowledge And Reasoning For Language Processing*, pages 42–49. Association for Computational Linguistics.
- Motro, A. (1994). Intensional Answers to Database Queries. *IEEE Transactions on Knowledge and Data Engineering*, 6(3):444–454.
- Naber, D. and Milkowski, M. (2016). LanguageTool Style and Grammar Check. Technical report, LanguageTool Organization.
- Nilagupta, S. (1977). The Relationship of Syntax to Readability for ESL Students in Thailand. *Journal of Reading*, 20(7):585–594.
- O’Neill, A. (2011). DictService: Word Dictionary Web Service. Technical report, DICT Service.
- Panther, K. and Thornburg, L. (2002). A Conceptual Analysis of English -er Nominals. In *Applied Cognitive Linguistics II: Language Pedagogy*, pages 280–319. Walter de Gruyter.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.
- Perera, R., Nand, P., and Naeem, A. (2017). Utilizing Typed Dependency Subtree Patterns for Answer Sentence Generation in Question Answering Systems. *Progress in Artificial Intelligence*, 6(2):105–119.
- Poesio, M., Stevenson, R., Eugenio, B. D., and Hitzeman, J. (2004). Centering: A Parametric Theory and Its Instantiations. *Computational Linguistics*, 30(3):309–363.
- Porter, M. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3):130–137.
- Porter, M. F. (2001). Snowball: A language for stemming algorithms. Technical report, Snowball Project.
- Quarteroni, S. (2010). Personalized Question Answering. *Traitement Automatique des Langues*, 51(1):97–123.

- Reiter, E. and Belz, A. (2009). An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems. *Computational Linguistics*, 35(4):529–558.
- Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, United Kingdom.
- Santos, J. R. A. (1999). Cronbach’s alpha: A Tool for Assessing the Reliability of Scales. *Journal of Extension*, 37(2):1–5.
- Schäfer, F. (2011). Naturally Atomic er-nominalizations. *Recherches linguistiques de Vincennes*, 40(1):27–42.
- Small, S., Liu, T., Shimizu, N., and Strzalkowski, T. (2003a). HITIQA: An Interactive Question Answering System: A Preliminary Report. In *ACL Workshop on Multilingual summarization and Question Answering*. Association for Computational Linguistics.
- Small, S., Shimizu, N., Strzalkowski, T., and Liu, T. (2003b). HITIQA: A Data Driven Approach to Interactive Question Answering: A Preliminary Report. In *AAAI Spring Symposium*. Association for the Advancement of Artificial Intelligence.
- Smith, E. a. and Senter, R. J. (1967). Automated Readability Index. Technical report, University of Cincinnati.
- Stribling, J., Krohn, M., and Aguayo, D. (2005). SCIGen - An Automatic CS Paper Generator. Technical report, Massachusetts Institute of Technology.
- Suchanek, F. M., Kasnec, G., and Weikum, G. (2007). YAGO : A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In *16th international conference on World Wide Web*, pages 697–706, Alberta. ACM Press.
- Thai, V., O’Riain, S., Davis, B., and O’Sullivan, D. (2006). Personalized Question Answering: A Use Case for Business Analysis. In *First International Conference on Applications and Business Aspects of the Semantic Web*, pages 61–73. CEUR-WS.
- Theune, M., Schooten, B. V., Akker, R. O. D., Bosma, W., Hofs, D., Nijholt, A., Krahmer, E., Hooijdonk, C. V., and Marsi, E. (2007). Questions, Pictures, Answers: Introducing Pictures in Question-Answering Systems. In *International Symposium on Social Communication*, pages 450–463, Santiago de Cuba.
- Unger, C., Cimiano, P., Lopez, V., Motta, E., Buitelaar, P., and Cyganiak, R. (2012). Question Answering over Linked Data (QALD-2). In *Workshop Interacting with Linked Data (ILD)*, Heraklion, Greece. CEUR-WS.
- Vargas-Vera, M. and Motta, E. (2004). AQUA-Ontology-based Question Answering System. In *Mexican International Conference on Artificial Intelligence*, Mexico City, Mexico. Springer-Verlag.
- von Glasersfeld, E. (1970). The Problem of Syntactic Complexity in Reading and Readability. *Journal of Literacy Research*, 3(2):1–14.

- Vrandečić, D. and Krötzsch, M. (2014). Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, 57(10):78–85.
- Walker, M. A., Whittaker, S. J., Stent, A., Maloor, P., Moore, J., Johnston, M., and Vasireddy, G. (2004). Generation and Evaluation of User Tailored Responses in Multimodal Dialogue. *Cognitive Science*, 28(5):811–840.
- Walter, S., Unger, C., and Cimiano, P. (2013). A Corpus-Based Approach for the Induction of Ontology Lexica. In *18th International Conference on Applications of Natural Language to Information Systems*, pages 102–113, Salford. Springer-Verlag.
- Wang, Y.-Y., Acero, A., and Chelba, C. (2003). Is Word Error Rate a Good Indicator for Spoken Language Understanding Accuracy. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 577–582. IEEE.
- Webber, B., Gardent, C., and Bos, J. (2002). Position Statement: Inference in Question Answering. In *3rd international conference on Language Resources and Evaluation (LREC)*. European Language Resources Association.
- Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. (1999). KEA: Practical Automatic Keyphrase Extraction. In *4th ACM Conference on Digital Libraries*, pages 254–261. ACM.
- Yu, H., Lee, M., Kaufman, D., Ely, J., Osheroff, J. A., Hripcsak, G., and Cimino, J. (2007). Development, Implementation, and a Cognitive Evaluation of a Definitional Question Answering System for Physicians. *Journal of Biomedical Informatics*, 40(3):236–251.
- Zhang, P., Wu, C., Wang, C., and Huang, X. (2006). Personalized Question Answering System Based on Ontology and Semantic Web. In *IEEE International Conference on Industrial Informatics*, pages 1046–1051. IEEE.

Appendix A

Sample Test Question Results

Which U.S. State has the abbreviation MN?

Answer

Minnesota

Informative Answer

Answer Sentence

Minnesota has the abbreviation MN.

Entity Descriptions

Minnesota is an administrative region. It is a state in United States. Saint Paul, Minnesota is the capital of it. Its area total, area land, and area water are respectively 225.181 billion m², 206 billion m², and 18.99 billion m². Its minimum elevation, and maximum elevation are respectively 183.0 m, and 701.0 m.

How many people live in the capital of Australia?

Answer

367752

Informative Answer

Answer Sentence

Three hundred sixty seven thousand seven hundred fifty two people live in the capital of Australia.

Entity Descriptions

Australia is a country. Its long name is Commonwealth of Australia. Canberra is the capital city of it. Largest city in it is Sydney. Australian dollar is the official currency of Australia. Its anthem is Advance Australia Fair. Its population density is 2.8ppkm². Australia's area total is 7692 billion m². Julia Gillard is the Prime Minister of it.

Who are the parents of the wife of Juan Carlos?

Answer

Frederica of Hanover and Paul of Greece

Informative Answer

Answer Sentence

The parents of the wife of Juan Carlos are Frederica of Hanover and Paul of Greece.

Entity Descriptions

Juan Carlos I of Spain is a royalty. His alias is Carlos Alfonso Victor Maria de Borbon y Borbon-Dos Sicilias. He is the son of Princess María de las Mercedes of Bourbon-Two Sicilies, and Infante Juan, Count of Barcelona. He was born on January 05, 1938 in Kingdom of Italy. Queen Sofía of Spain married him. He is preceded by Alejandro Rodríguez de Valcárcel. Felipe VI of Spain succeeded him.

Frederica of Hanover was a royalty. She was the daughter of Ernest Augustus, Duke of Brunswick, and Princess Victoria Louise of Prussia. She was born on April 18, 1917 in Duchy of Brunswick. Frederica died on February 06, 1981 in Spain. Paul of Greece married her.

Paul of Greece was a royalty. He was the son of Constantine I of Greece, and Sophia of Prussia. He was born on December 14, 1901 in Kingdom of Greece. Paul died on March 06, 1964 in Kingdom of Greece. Frederica of Hanover married him. He is preceded by George II of Greece. Constantine II of Greece succeeded him.

Who created the comic Captain America?**Answer**

Jack Kirby and Joe Simon

Informative Answer**Answer Sentence**

Jack Kirby and Joe Simon created the comic Captain America.

Entity Descriptions

Captain America is a comics character. It was created by Jack Kirby, and Joe Simon.

Jack Kirby was a comics creator. He was an American book artist. He was born on August 28, 1917 in New York City. Jack died on February 06, 1994 in Thousand Oaks, California.

Joe Simon was a comics creator. He was born on October 11, 1913 in New York. He won the Eisner Award, and the Inkpot Award. Joe died on December 14, 2011 in New York City.

In which U.S. state is Area 51 located?**Answer**

Nevada

Informative Answer**Answer Sentence**

Area 51 is located in Nevada.

Entity Descriptions

Area 51 is an airport. The airport is located in Southern Nevada desert. It is owned by Federal government of the United States. Its elevation is 1360.02 m.

United States is a country. Its long name is United States of America. Washington, D.C. is the capital city of it. Largest city in it is New York City. United States dollar is the official currency of United States. Its anthem, and motto are respectively The Star-Spangled Banner, and "In God we trust". Its population density is 34.2ppkm².

United States's area total is 9.83 trillion m². John Roberts is the Prime Minister of it. It has influenced American music.

Nevada is an administrative region. It is a state in United States. Carson City is the capital of it. De jure, and De facto are official languages of the state of Nevada. Its area total, area land, and area water are respectively 286367 million m², 284 billion m², and 1.971 billion m². Its minimum elevation, and maximum elevation are respectively 147.0 m, and 4007.1 m.

How tall is Michael Jordan?

Answer

1.9812

Informative Answer

Answer Sentence

Michael Jordan is 1.9812m tall.

Entity Descriptions

Michael Jordan is a basketball player. He was born on February 16, 1963 in New York. His weight, and height are respectively 97.98 kg, and 1.98 m. He attended Emsley A. Laney High School. His number is 231245. He retired in 2003. He was a Shooting guard. His term periods are January 01, 1984 to January 01, 1984, and January 01, 2001 to January 01, 2002.

What is the birth name of Angela Merkel?

Answer

Angela Dorothea Kasner

Informative Answer

Answer Sentence

The birth name of Angela Merkel is Angela Dorothea Kasner.

Entity Descriptions

Angela Merkel is an office holder. She was born on July 17, 1954 in Hamburg. She is known as Angela Dorothea Kasner. Angela attended Leipzig University. She was the first president from the Christian Democratic Union , and the Democratic Awakening. She was the Chancellor of Germany, the Minister of Women and Youth, and the Minister of the Environment, among others. Her term periods are January 18, 1991 to November 17, 1994, and November 17, 1994 to October 26, 1998. Claudia Nolte succeeded her.

Was Margaret Thatcher a chemist?

Answer

True

Informative Answer

Answer Sentence

Margaret Thatcher was a chemist.

Entity Descriptions

Margaret Thatcher was an office holder. She was a chemist, and a lawyer. She was born on October 13, 1925 in Grantham. Margaret is known as Margaret Hilda Roberts. Her alias was Roberts, Margaret Hilda. She studied at Somerville College, Oxford, and City Law School. Her husband was Denis Thatcher. Her children were Carol Thatcher, and Mark Thatcher. She was a member of Conservative Party. She was the for Finchley, the Secretary of State for Education and Science, and the Leader of the Conservative Party, among others. Margaret died on April 08, 2013 in London. Her term periods were March 05, 1974 to February 11, 1975, June 20, 1970 to March 04, 1974, and February 11, 1975 to November 28, 1990, among others. Norman Pentland succeeded her.

Who founded Intel?**Answer**

Gordon Moore and Robert Noyce

Informative Answer**Answer Sentence**

Gordon Moore and Robert Noyce founded Intel.

Entity Descriptions

Intel is a company. The company was founded on July 18, 1968. It is founded by Robert Noyce, and Gordon Moore. It is located in Santa Clara, California. Its equity is \$58.2 billion. Its net income, operating income, and revenue are respectively \$9620 million,

\$12 billion, and \$52.7 billion. It produces Motherboard, Flash memory, and Bluetooth, among others. It employs 107600 employees.

Gordon Moore is a scientist. His birth name is Gordon Earle Moore. His alias is Moore, Gordon Earle. He was born on January 03, 1929 in San Francisco. He co-founded Gordon and Betty Moore Foundation Corporation, Moore's law Corporation, and Intel Corporation. Gordon graduated from San Jose State University, University of California, Berkeley, and California Institute of Technology. He is the recipient of the National Medal of Technology and Innovation, the IEEE Medal of Honor, and the Presidential Medal of Freedom.

Robert Noyce worked as an Intel co-founder of Fairchild Semiconductor. He was born on December 12, 1927 in Burlington. His alias was Noyce, Bob. He graduated from Grinnell College, and Massachusetts Institute of Technology. Elizabeth Noyce married him. He died on June 03, 1990 in Texas.

What is the time zone of Salt Lake City?

Answer

Mountain Time Zone

Informative Answer

Answer Sentence

The time zone of Salt Lake City is Mountain Time Zone.

Entity Descriptions

Salt Lake City is a settlement. It is the capital of the United States. It is part of Salt Lake County, Utah, and Utah. Its leader name is Ralph Becker. Its area total, area land, and area water are respectively 286 million m², 283 million m², and 3.3 million m². Salt Lake City's area code is 385, 801. Its elevation is 1288.0 m. Its population total, and population urban are respectively 189314, and 2.35 million. Salt Lake City's population density is 643.3ppkm². Its population metro is 1.15 million. Its time zone is Mountain Time Zone.

Who developed Skype?**Answer**

Skype Technologies and Microsoft

Informative Answer**Answer Sentence**

Skype Technologies and Microsoft developed Skype.

Entity Descriptions

Skype is a software. It supports Videoconferencing, Instant messaging, and Voice over IP. It was developed by Janus Friis, and Niklas Zennström. Skype was developed by Microsoft, and Skype Technologies. It is released under the Freemium. It is available for Symbian, Android , and Microsoft Windows, among others. Skype is written in Object Pascal, C , and Objective-C.

Skype Technologies is a company. The company is located in Luxembourg. Its revenue is \$1 billion. It produces Voice over IP, and Skype. It employs 500 employees.

Microsoft is a company. The company was founded on April 04, 1975 in 1975. It is founded by Paul Allen, and Bill Gates. It was founded in New Mexico, Albuquerque, New Mexico, and United States. Microsoft is located in Microsoft Redmond Campus, and Redmond, Washington. It acquired List of mergers and acquisitions by Microsoft. Its equity is \$78.9 billion. Its net income, operating income, and revenue are respectively \$21860 million, \$26.76 billion, and \$77850 million. It released Microsoft Windows. It employs 101914 employees.

Who is the husband of Amanda Palmer?

Answer

Neil Gaiman

Informative Answer

Answer Sentence

The husband of Amanda Palmer is Neil Gaiman.

Entity Descriptions

Amanda Palmer is a musical artist. She is known as Amanda F. Palmer. She was born on April 30, 1976 in New York City. Evelyn Evelyn, Theatre Is Evil, and 8in8, among others, are duo formed by her. She signed with Roadrunner Records.

Neil Gaiman is a writer. His birth name is Neil Richard Gaiman. He worked as a Writer. He was born on November 10, 1960 in Portchester. Neil is married to Amanda

Palmer. He is influenced by Alan Moore, Ray Bradbury, and Ursula K. Le Guin, among others. He has written a direct sequel to *Coraline*, *The Graveyard Book*, and *American Gods*, among others.

Index

AI, 1, 218
ARI, 51
AVM, 61
CFG, 85, 97, 101, 153, 165
ClosedIE, 106
CNL, 97
DBpedia, 5, 61, 64, 66
ILP, 36
IPA, 213
IT, 26
LCS, 34, 54
Lexicalization, 92
LGG, 36
Linked Data, 4
MT, 53, 203
NLG, 5, 137, 211
NLP, 50, 212
ODF, 138
OpenIE, 102, 106, 107, 124
OWL, 65, 71
POJO, 61
POM, 107
POS, 83
QA, 2, 13, 82, 83, 204
QALD, 4, 62, 66
RDF, 62, 69, 117, 160
RealText, 4
REG, 126
RST, 16, 17
SPARQL, 4, 5, 69, 76, 79, 91
SSML, 138
TFIDF, 19
WER, 54
WWW, 210