

# Interaction history based answer formulation for question answering

Rivindu Perera and Parma Nand

School of Computer and Mathematical Sciences  
Auckland University of Technology  
Auckland, New Zealand  
`{rivindu.perera,parma.nand}@aut.ac.nz`

**Abstract.** With the rapid growth in information access methodologies, question answering has drawn considerable attention among others. Though question answering has emerged as an interesting new research domain, still it is vastly concentrated on question processing and answer extraction approaches. Latter steps like answer ranking, formulation and presentations are not treated in depth. Weakness we found in this arena is that answers that a particular user has acquired are not considered, when processing new questions. As a result, current systems are not capable of linking two questions such as, "When is the Apple founded?" with a previously processed question "When is the Microsoft founded?" generating an answer in the form of "Apple is founded one year later Microsoft founded, in 1976". In this paper we present an approach towards question answering to devise an answer based on the questions already processed by the system for a particular user which is termed as interaction history for the user. Our approach is a combination of question processing, relation extraction and knowledge representation with inference models. During the process we primarily focus on acquiring knowledge and building up a scalable user model to formulate future answers based on current answers that same user has processed. According to evaluation we carried out based on the TREC resources shows that proposed technology is promising and effective in question answering.

**Keywords:** Question answering, Answer formulation, Interaction history, Natural Language Processing

## 1 Introduction

Question answering systems are designed to present an answer for a given question composed in natural language. Due to this fact, from existing question answering systems such as AnswerBus [1], START [2] and WolframAlpha [3] to early question answering systems like LUNAR and BASEBALL [4] shared the same objective of generating the answer through diverse research attempts. All aforementioned question answering systems considered questions as independent units and generated answers based on the information retrieval and extraction

modules that are integrated. Recently, when attempting to investigate new dimensions in question answering, answer formulation and presentation also became a featuring factor [5].

Answer formulation is considered to be the final step in question answering systems which is responsible for presenting the answer to the user. As most of the early researches extensively focused on question processing and candidate answer extraction approaches, this unit kept relatively untouched. However, competition which is arisen among question answering systems, opened the path to discover novel ways of answer formulation which can immensely contribute for the uniqueness of a question answering system.

In this paper, we investigate a technique called Interaction History based Answer Formulation (IHAF) to build a user model to generate more personalized answers for users based on the past interactions for that particular user. We conduct experiments based on TREC question set and with different variations. However, it should be emphasised that in this implementation of question answering system, we are not going to mine the answers from the web or from a corpus. Instead we have provided the ranked sentence list from TREC to extract answers. This is carried out basically because of our attention is not on question answering itself but on answer formulation where underlying question answering system is not relevant for the process.

The remainder of the paper is organized as follows. The next section is committed to related work. In Section 3, we discuss about our approach towards the issue we have identified and provide complete overview of the technique. Section 4 is concentrated on results and evaluation schema we have used to assess our novel approach. In Section 5, discussion on the technique in relation to results is presented covering different perspectives. Finally, in Section 6, we conclude showing future directions of this new method towards question answering.

## 2 Related Work

Answer formulation for question answering systems is first brought into broad discussion by Hirschman and Gaizauskas [6] through an analysis of current question answering strategies. However, there are significant research attempts taken towards presenting and formulating an answer in question answering.

Wang et al. [7] present an interesting idea of generating answer based on relations mined based on focus information. Though, it does not express personalized answer generation method, employing pre-mined relations to generate answers draws the attention as a new method. Moving to an innovative direction authors of [8] propose a slightly different method. They utilize past answers to generate answers for new questions where they employ Yahoo! Answers track. This research is based on surface level feature extraction of top candidate and then classification through a random forest classifier. As answer generation is applied with the vision of social engagement, personalization of a particular answer based on the user is not treated. Evaluation shows that it has achieved good result in answer generation, but content applicability for the new question is not

measured in a qualitative approach. Similar technique used in [8] is noticed in [9] and [10] as well with slight changes in the process.

Ni et al. [11] build a topic based user interest model to integrate with question answering. Though this research sounds well in the area of user based question answering, the model they propose is used only to recommend appropriate questions. Also it is observed that this proposed model is using latent topic model which limit the model to top level topic based user model. In contrast to this method, seminal work presented by Nyberg et al. [12] show an approach inspired by methodology showcased by Hovy [13]. Utilization of user discourse to boost the question answering based on the classified question types signals that this research is focusing on the question processing based on the user supplied concepts. But the lack of user centred question processing is still not achieved efficiently here. Advanced approach towards question processing with existing knowledge is demonstrated in system proposed by Harabagiu and Hickl [14]. In this novel attempt, Harabagiu and Hickl endeavour the gap between existing knowledge utilization and answer formulation through a promising application. Though this research significantly lies in the shallow text processing area, the concept unveiled and model built are extremely useful when mining answers for questions based on the existing knowledge that is already acquired.

Moving few steps further, Higashinaka and Isozaki [15] encompass the usage of casual expression patterns for answer extraction. Grounded on a corpus based technique to mine features to build the casual expression patterns, researchers attempt to apply the resulting model in a open domain question answering system. Different dimensions in the evaluation carried out by this research shows that their model outperforms in most of the scenarios. Basically, employing a WHY-type question set in the training and attempting to build a typological answer extraction model seems to be promising. Several early attempts such as [16], [17] and [18] also express similar candidate answer formulation models. When it comes to the feature selection and applicability in the question processing steps, model developed by Higashinaka and Isozaki can be considered as a more improved procedure.

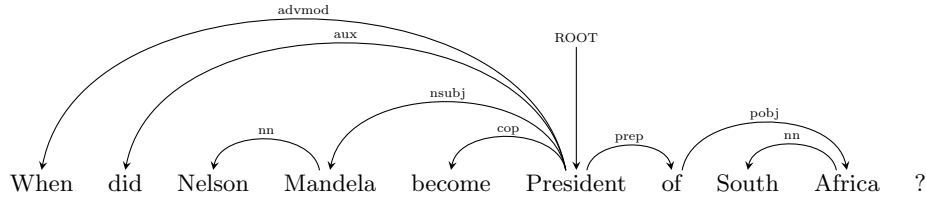
### 3 Interaction History based Answer Formulation (IHAF)

We have separated the contexts of user model building and answer formulation to two sections in order to focus on them more thoroughly. In next sections we delineate the framework of complete process resulted through our research. Before moving into the complete process, we first examine basic utilities that are built to serve the high level goal of the system.

#### 3.1 Basic utilities incorporated

In this section we define some basic functionalities that we employ during user model building and answer formulation.

**Generating Typed Dependency Parses** Typed dependency parsing is a way of representing dependencies between words given in a sentence structure with labelled grammatical relations [19] [20]. We employed dependency parsing mainly because of compared to a phrase structure with constituency, dependency parsing can easily transformed in to relation schema. Another, noteworthy point to notice here is that as labelled grammatical relations are present it can later be modified and normalized into more accurate relation if resulting typed dependency is not suitable to generate a relation. For the dependency parsing we utilize the Stanford parser based on 50 grammatical relations [20] [19]. Dependency Parse (DP) generated for a selected question from TREC development set is shown in Fig. 1.



**Fig. 1.** Example of dependency parse of a question

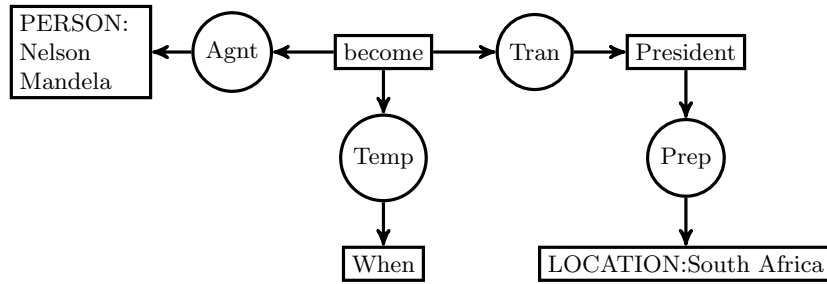
**From Dependency Parses to Conceptual Graph** Conversion from typed dependency parse to the Conceptual Graph (CG) is also considered as a principal process in representation workflow. Conceptual graphs introduced by Sowa [21] is considered as the basis of ontology and related knowledge representation strategies. An example CG we expect from question considered in Fig. 1 is shown in Fig. 2. As conversion of a language structure to a knowledge structure requires considerable effort, we have defined a set of rule set for 8 different case relations based on thematic roles, named entities and user defined types which are shown in Table 1. In all other cases where we are unable to label with appropriate case role of CG relation, by default typed dependency label is used.

We employed conceptual graphs after consideration several representation strategies such as lexical ontologies [22], frames [23] and semi-structured language fragments [24]. Factor that influenced us to choose CG, is the opportunity it provides to infer based on the model developed using Common Logic (CL) and six canonical formation rules [21] which supports us to perform projection operators which will be discussed in following section.

Key factor we noticed during preliminary analysis on conversion process is that DP structure is not identically mapped with CG. Therefore, simple conversion from one to one mapping converting DP root to CG root can lead us to severe errors in structure. Due to this major emphasis is placed on identifying a gener-

**Table 1.** Defined case relations

Case Relation	Abbreviation	Description
Agent	Agnt	Main actor of the event
Transition	Tran	Transition from one state to another
Attribute	Attr	Property of a object
Patient	Ptnt	Object which is subjected to an action
Experiencer	Expr	Object that experience the event
Recipient	Rcpt	Object that receive something
Temporal	Temp	Time related factor
State	Stat	Current state of the object

**Fig. 2.** Example of conceptual graph of a question

alizable rule set to convert DP structure to a CG representation. Algorithm. 1 depicts the flow of applying rules during the conversion phase.

For the phrase extraction, we employ a three step process of *a)* stop word removal of the question *b)* stemming using Porter stemmer [25] *c)* calculation of Term Frequency and Inverse Sentence Frequency (TF-ISF) for the decomposed question terms.

As an initial step, Named Entity Recognition (NER) is also engaged using Conditional Random Field based NER tool - Stanford NER. The idea of NER is to output more semantically rich CG constructs labelled with entity type, which is again useful in searching based on type. Following initial steps, we then check the best candidate for the root of the CG ( $CG_{Root}$ ). This is based on the condition that if DP is consisted of a copular verb ( $DP_{cop_v}$ ), then it is selected by default and if not we continue with root of DP ( $DP_{Root}$ ). Then after removal of all WH-pronouns, root of CG, root of DP and remaining auxiliary verbs and prepositions, we start relation generation where 9 different relation types are considered based on 9 different rule sets ( $ruleSet_{<TYPE>}$ ). Each relation is labelled with the matching relation found ( $<TYPE>_p$ ) based on the considered phrase ( $p$ ). In case if matching relation is not found based on the predefined rule set, relation present in the DP is used to label the generated relation. Result of this flow is a CG which we later can be used with projection operators.

---

**Algorithm 1:** Flow of converting DP to mapping CG

---

**Data:** DP: Typed dependency representation, P:Extracted phrases of question**Result:** CG:Conceptual Graph**begin**

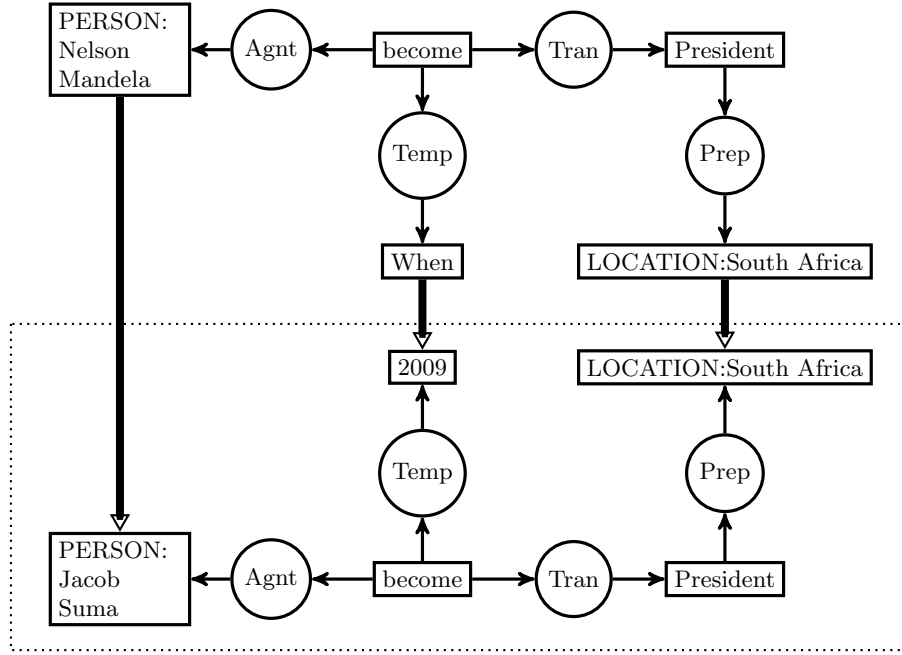
```

     $E \leftarrow RecognizeNamedEntities(Q);$ 
    if  $DP_{cop_v} \neq NULL$  and  $DP_{Root} \neq DP_{cop_v}$  then
         $CG_{Root} \leftarrow DP_{cop_v};$ 
    else
         $CG_{Root} \leftarrow DP_{Root};$ 
     $P \leftarrow P - \{WH_{pronoun}, CG_{Root}, DP_{prep}, DP_{aux}\};$ 
    for  $p \in P$  do
        switch  $p$  do
            // Only one case is shown for 8 cases needed for 8
            // different relation types - replacing <TYPE> generic
            // type with Agnt, Ptnt, Expr, etc.
            case  $relation_p \in ruleSet_{<TYPE>}$ 
                if  $p$  hasRelationWith  $CG_{Root}$  then
                     $\perp generateRelation(<TYPE>_p, CG_{Root});$ 
                else
                     $\perp generateRelation(<TYPE>_p, DP_{relatedPhrase});$ 
            // ..... remaining cases
            otherwise
                 $\perp generateRelation(DP - Type_p, DP_{relatedPhrase});$ 
        if  $p \in E$  then
             $\perp labelWithEntityType(p);$ 

```

---

**Conceptual Graph Projection** CG projection is the process of extracting or generating advanced knowledge based on two or more CGs. In our case, we utilize a simple projection operator based on nodes in the CG. Properties that are considered in projection are, named entities, root of CG and the specified CG relation (ex: agent, patient, etc.). Example projection based on our previous example in Fig. 2 is shown in Fig. 3. In this example we assume that CG for "Jacob Suma became president of South Africa in 2009" exists in the knowledge base.



**Fig. 3.** Example of conceptual graph projection

According to the projection shown in Fig. 3, it can be noticed that named entities and temporal event annotations (When and 2009) are projected providing opportunity to extract necessary knowledge. Building the required natural language answer will be discussed in Section 3.3.

### 3.2 Building the user model

Most significant element that is presented by this research can be considered as the user model that we have designed to build in order to link questions with previously processed questions by the same user. However, it is heavily dependant on basic utilities we described in Section 3.1.

Below we have defined the basic work flow of the user model creation based on the utilities discussed above.

1. Generate the Typed Dependency Parses (DP) of the question
2. Map the parsed question to Conceptual Graph (CG)
3. Save the CG to the knowledge base

The knowledge base constructed at this point will be unique and will be considered as the user model for that particular user.

For the initial training 120 questions are suggested to commence the model building process as it needs considerable amount of question classes and answer candidates to link with next questions. This 120 questions, defined as the start-up items are selected based on 3 factors from TREC-9 data set, *a*) question type (WH-type) *b*) question target (person, location, date/time, etc.) *c*) named entities mentioned (ex: South Africa, Nelson Mandela, Jacob Suma, etc). Basis of selecting these 3 factors is adopted from several past research attempts which have highlighted the importance of them running empirical methods towards question answering [26–28].

### 3.3 Answer formulation

In this section, we describe the approach taken towards generating answer for a given question employing an existing user model.

1. Generate the Typed Dependency Parses (DP) of the question
2. Map the parsed question to Conceptual Graph (CG)
3. Project CG to user model elements and extract similar structures
4. Generate language structures based on inferences and previous interactions acquired from user model

Answer formulation also utilize the basic utilities discussed, but move further with extracted knowledge from user model by building natural language. For the purpose of building naturally looking answer, we employ a surface realization engine where we get the final answer based on the structures defined. SimpleNLG [29] has shown high flexibility during our analysis for suitable realization engines. Structuring templates are defined by authors for the SimpleNLG to output the final natural language text for the given answer.

## 4 Results

For the evaluation, we employed TREC-8 [30] question set containing 200 questions. Main reason behind this section is that unlike TREC-9 and later TREC data sets, TREC-9 specifically focus on entity based answer extraction. As the approach discussed here is also based on entity types, TREC-8 provided us the most appropriate selection for the context.

To get a better understanding of the process, we carried out the evaluation in



**Table 2.** Evaluation setup

Type	Traning setup (#questions [TREC-type])	Testing setup (#questions [TREC-type])
T-1	120 [TREC-9]	200 [TREC-8]
T-2	240 [TREC-9]	200 [TREC-8]
T-3	360 [TREC-9]	200 [TREC-8]
T-4	480 [TREC-9]	200 [TREC-8]
T-5	200 [TREC-8]	200 [TREC-8]

5 different steps which is depicted in Table 2 with different number of questions for both training and testing.

During evaluation, we noticed that user model built based on initial training phase is used for four different target types, date/time, number, currency and distance are not fairly distributed in two different TREC datasets. It is also valuable to mention that while the execution, our system was able to link questions from testing set as well , but which we neglected as we strictly believed that answers must be drawn only from training set. To provide a baseline for the evaluation, we consulted Cosine Similarity (CS) between new question and previously processed question. If two questions (stop words removed) are shown in vectors  $A$  and  $B$ , then CS between  $A$  and  $B$  can be calculated as,

$$CS(A, B) = \frac{\sum_{i=1}^n t_{Ai} \times t_{Bi}}{\sqrt{\sum_{i=1}^n (t_{Ai})^2} \times \sqrt{\sum_{i=1}^n (t_{Bi})^2}} \quad (1)$$

where,  $t_i$  is the term weight for a word  $w_i$ , for which TF-ISF (Term Frequency, Inverse Sentence Frequency) value is used.

**Table 3.** Evaluation results

Test type	Precision	Recall	F-measure (IHBQA)	F-measure (baseline)
T-1	0.38	0.5	0.43	0.34
T-2	0.53	0.6	0.56	0.42
T-3	0.42	0.53	0.46	0.41
T-4	0.39	0.52	0.44	0.29
T-5	0.53	0.8	0.63	0.55

Table 3 shows the results we acquired during our evaluation phase. While testing with TREC-8 question sets, we noticed that some questions exists with answers having different measurement units. For example, miles and kilometres both are used (ex: Question 9 and Question 150) when providing answers for questions. To provide equal treatment during answer search, we manually converted them to metric units. It is also noteworthy to mention that as we are

not involving with a question processing and answer extraction unit as in most question answering systems, results we achieved are completely focused on answer formulation. Therefore, error rate occurs in actual answer extraction with incorrect answers is not included here, as we are already having the exact answer for every question processed.

## 5 Discussion

According to the evaluation we carried out in 5 stage process, it can be noticed that random incrementation of number of question has no effect in accuracy of answer formulation. Through examination of this issue, highlighted two main factors to further investigate. Firstly, as we are incrementing questions in random fashion, number of recognized named entities has not proportionally incremented. This is partially due to the weakness of the NER tool that we employed to this research. Secondly, when number of questions increase, it is observed that CG projection operator has more choices based on inference rules used. This has lead the system to retrieve some of the less significant items due to overlapping inference rules.

Generally, our approach has produced errors when processing context sensitive questions which are difficult to determine the type by WH-pronoun. One such erroneous observation is following two questions which are linked during evaluation without having similar context or target type:

- What is the fare cost for the round trip between New York and London on Concorde?
- What is the duration of the trip from Bristol to London by rail?

During the evaluation, it has reached to its highest accuracy in test type-5, when same test collection is used for the training phase. As we investigated this is mainly due to the style of TREC-8 questions, which are more focused on entity types and much simpler than TREC-9 questions considering context and target identification.

Though these experiments provides standard levels to compare with, the most appropriate evaluation that can be applied for a question answering system in this nature is the real world evaluation. As there are multiple entities incorporated and question types are difficult to determine, answer formulation for real world question answering can place a benchmark on evaluation in a more precise manner. However, at this stage, no real world evaluation is done with human involvement which lies as a future goal.

Overall, though our approach has not achieved high accuracy in evaluation, it can be noticed that approach is still feasible and promising in answer formulation. As mentioned earlier, observed issues are mainly due to the process we have engaged for the CG construction and processing which can be considered as a main unit that contributes for user model building.

## 6 Conclusion and future work

We presented a novel, answer formulation approach for question answering based on the concept of interaction history of a user. In contrast to previous approaches, this new model targeted on more personalized answer - considering that linking with earlier acquired knowledge can dramatically increase the usefulness of an answer. User model and CG generation methods presented through this research can be seen as first step towards answer formulation for question answering with emphasis on interaction history. However, it should not be seen as a complete model. Through, observations during the evaluation, we noticed severe issues that lead our system to erroneous results as mentioned in previous section. Addressing these issues is necessary and important when attempting to build more personalized answer formulation for open domain question answering.

## References

1. Zheng, Z.: AnswerBus question answering system. In: Second International Conference on Human Language Technology Research, Morgan Kaufmann Publishers Inc. (March 2002) 399–404
2. Katz, B., Borchardt, G., Felshin, S.: Natural Language Annotations for Question Answering\*. In: 19th International FLAIRS Conference, Melbourne Beach, Florida (2006)
3. Research, W.: WolframAlpha API (2011)
4. Woods, W.A.: Semantics and quantification in natural language question answering. (June 1986) 205–248
5. Maybury, M.: New Directions In Question Answering. In Strzalkowski, T., Harabagiu, S.M., eds.: Advances in Open Domain Question Answering. Volume 32 of Text, Speech and Language Technology. Springer Netherlands, Dordrecht (2008)
6. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. *Natural Language Engineering* **7**(04) (February 2002) 275–300
7. Wang, R., Ning, C., Lu, B., Huang, X.: Research on answer generation method based on focus information extraction. In: 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE). Volume 2., IEEE (May 2012) 724–728
8. Shtok, A., Dror, G., Maarek, Y., Szpektor, I.: Learning from the past. In: Proceedings of the 21st international conference on World Wide Web - WWW '12, New York, New York, USA, ACM Press (April 2012) 759
9. Wang, G., Gill, K., Mohanlal, M., Zheng, H., Zhao, B.Y.: Wisdom in the social crowd: an analysis of quora. (May 2013) 1341–1352
10. Liu, D.R., Chen, Y.H., Kao, W.C., Wang, H.W.: Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management* **49**(1) (January 2013) 312–329
11. Ni, X., Lu, Y., Quan, X., Wenyan, L., Hua, B.: User interest modeling and its application for question recommendation in user-interactive question answering systems. *Information Processing & Management* **48**(2) (March 2012) 218–233
12. Nyberg, E., Frederking, R.: JAVELIN. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology Demonstrations - NAACL '03. Volume 4., Morristown, NJ, USA, Association for Computational Linguistics (May 2003) 19–20

13. Hovy, E., Hermjakob, U., Ravichandran, D.: A question/answer typology with surface text patterns. (March 2002) 247–251
14. Harabagiu, S., Hickl, A.: Using scenario knowledge in automatic question answering. (July 2006) 32–39
15. Higashinaka, R., Isozaki, H.: Automatically Acquiring Causal Expression Patterns from Relation-annotated Corpora to Improve Question Answering for why-Questions. *ACM Transactions on Asian Language Information Processing* **7**(2) (June 2008) 1–29
16. Yang, F., Feng, J., Di Fabbri, G.: A data driven approach to relevancy recognition for contextual question answering. (June 2006) 33–40
17. Li, B., Jin, T., Lyu, M.R., King, I., Mak, B.: Analyzing and predicting question quality in community question answering services. In: *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*, New York, New York, USA, ACM Press (April 2012) 775
18. Stoyanchev, S., Song, Y.C., Lahti, W.: Exact phrases in information retrieval for question answering. (August 2008) 9–16
19. Marie-Catherine de Marneffe, Bill MacCartney, C.D.M.: Generating typed dependency parses from phrase structure parses. In: *International conference on language resources and evaluation*. (2006)
20. de Marneffe, M.C., Manning, C.D.: The Stanford typed dependencies representation. In: *Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, Association for Computational Linguistics (August 2008) 1–8
21. Sowa, J.F.: Conceptual graphs as a universal knowledge representation. *Computers & Mathematics with Applications* **23**(2-5) (January 1992) 75–93
22. Veale, T., Hao, Y.: A context-sensitive framework for lexical ontologies. *The Knowledge Engineering Review* **23**(01) (March 2008) 101–115
23. Martin, P.: Knowledge Representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. (July 2002) 77–91
24. Liu, H., Singh, P.: ConceptNet A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal* **22**(4) (October 2004) 211–226
25. Porter, M.: An algorithm for suffix stripping. *Program* **14**(3) (1980) 130–137
26. Hartrumpf, S.: Adapting a semantic question answering system to the web. (April 2006) 61–68
27. Grappy, A., Grau, B., Rosset, S.: Methods combination and ML-based re-ranking of multiple hypothesis for question-answering systems. (April 2012) 87–96
28. Clarke, C.L.A., Terra, E.L.: Approximating the top-m passages in a parallel question answering system. In: *Proceedings of the Thirteenth ACM conference on Information and knowledge management - CIKM '04*, New York, New York, USA, ACM Press (November 2004) 454
29. Gatt, A., Reiter, E.: SimpleNLG: a realisation engine for practical applications. (March 2009) 90–93
30. Voorhees, E.: Overview of TREC 1999. In: *TREC-8*, Gaithersburg, National Institute of Standards and Technology (1999)