Question-Answering with Relaxed Unification

Tony Abou-Assaleh Nick Cercone Vlado Kešelj
Faculty of Computer Science, Dalhousie University,
Halifax, Nova Scotia, Canada
{taa,nick,vlado}@cs.dal.ca

Abstract

Classical unification is strict in the sense that it requires a perfect agreement between the terms being unified. In practise, data are seldom error-free and can contain incorrect information. Classical unification fails when the data are imperfect. Relaxed unification is a new formalism that relaxes the rigid constraints of classical unification and enables reasoning under uncertainty and in the presence of inconsistent data. We illustrate how relaxed unification can be employed in a question-answering system, describe the system architecture, and propose the TREC Question-Ansewr Track to be an objective testbed of our approach.

1 Introduction

The classical unification function [9, 11] takes two terms as input and produces a boolean value indicating whether the unification succeeds. In the case of a result of true, the function also returns a substitution that unifies these two terms. The unification fails if the same feature is assigned different values in the objects being unified. This process places rigid constraints on the data requiring it to be correct and consistent. Since real-world data is seldom perfect, the classical unification fails at the encounter of the slightest error. Erroneous data often contains enough information that one can exploit to overcome the errors. In other cases, it is possible to draw approximate or less certain conclusions.

Relaxed unification [1, 2] provides a method for extracting information from imperfect data. To achieve this functionality, we relax the constraint that the values being unified must be identical. Instead, each value is replaced with a set containing the value as an element. Unifying two sets containing different values results in a new set containing the values from both sets. Since relaxed unification always succeeds, an evaluation function is needed to compute the degree of the mismatch in terms.

When analyzing natural languages in real-world applications, using formalisms that rely on the classical unification mechanism may yield unsatisfactory results. Text collected from the World Wide Web may contain typing errors, linguistic errors, or simply contain incorrect or incomplete information. Even when the text is error-free, unification-based grammars are imperfect. For instance, the English lexicons used in HPSG represent only a subset of English words and their meanings. To overcome these limitations in the question-answering task, we propose a framework for Head-driven Phrase Structure Grammar (HPSG) [10] based on relaxed unification. By substituting the classical unification mechanism in HPSG with the relaxed-unification mechanism and defining a suitable evaluation function, one is able to answer questions that are traditionally difficult to answer.

Section 2 illustrates through an example how relaxed unification can help answer questions that HPSG with classical unification fails to answer. We present a framework for a question-answering system based on HPSG and relaxed unification in section 3. Section 4 discusses an objective testbed for evaluating our proposed system. We conclude with a summary and future work directions in section 5.

2 Question Answering with Relaxed Unification

Kešelj [8] proposed that unification-based grammars, such as HPSG, can be used in question answering and showed how an answer can be extracted from passages when an attribute-value matrix (AVM) representing the question and an AVM representing a passage unify. Such an exact match is not always possible. For instance, consider the question

"Who does Alice like?"

and the two passages

1. Alice Loves Bob.

2. Alice hates Trudy.

AVM 1 represents the relevant parts of the question AVM.

$$\begin{bmatrix} \textit{question} & \\ \text{SELECT:} & \boxed{1} \\ \text{WHERE:} & \begin{bmatrix} \textit{top} & \\ \text{SEM:} & \begin{bmatrix} \textit{action} & \\ \text{EVENT:} & \text{like} \\ \text{SUBJ:} & \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Alice'} \end{bmatrix} \\ \text{OBJ:} & \boxed{1} \end{bmatrix} \end{bmatrix} \\ \text{QCAT:} & \text{WHO} \end{bmatrix} \tag{1}$$

AVMs 2 and 3 represent the relevant parts of the passage 1 and 2 AVMs, respectively.

$$\begin{bmatrix} \textit{vp} & \\ \text{SEM:} & \begin{bmatrix} \textit{action} & \\ \text{EVENT:} & \text{love} \\ \text{SUBJ:} & \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Alice'} \end{bmatrix} \\ \text{OBJ:} & \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Bob'} \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} \textit{vp} & \\ \text{SEM:} & \begin{bmatrix} \textit{action} & \\ \text{EVENT:} & \text{hate} \\ \text{SUBJ:} & \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Alice'} \end{bmatrix} \\ \text{OBJ:} & \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Trudy'} \end{bmatrix} \end{bmatrix} \end{bmatrix} \tag{3}$$

Each of AVM 2 and 3 gives a different value for EVENT than does AVM 1. Therefore, the unification of either of the passage AVMs with the question AVM fails. Using HPSG in such cases is not advantageous; the passage AVMs are treated as equal mismatches with the question AVM. This behaviour is correct and desired in a system that looks for an exact match and where the verbs 'like', 'love', and 'hate' carry unique meanings. However, such preciseness in meaning is often undesirable. Ambiguity in the language, and the fact that one word may denote several meanings, call for a more relaxed treatment of words.

Relax unifying AVM 2 with AVM 1 results in AVM 4. The mismatch in the value of EVENT is captured in the AVM.

$$\begin{bmatrix} \textit{question} & \\ \text{SELECT:} & \boxed{1} \\ \text{WHERE:} & \begin{bmatrix} \textit{top} & \\ \text{SEM:} & \begin{bmatrix} \textit{action} & \\ \text{EVENT:} & \{\text{like, love}\} \\ \text{SUBJ:} & \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Alice'} \end{bmatrix} \\ \text{OBJ:} & \boxed{1} \begin{bmatrix} \textit{np} & \\ \text{ORTH:} & \text{`Bob'} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{QCAT:} & \text{WHO} \end{bmatrix} \tag{4}$$

Similarly, relax unifying AVM 3 with AVM 1 results in AVM 5.

$$\begin{bmatrix} \textit{question} & \\ \text{SELECT:} & \boxed{1} \\ \text{WHERE:} & \begin{bmatrix} \textit{top} & \\ \text{SEM:} & \begin{bmatrix} \textit{action} & \\ \text{EVENT:} & \{\text{like, hate}\} \\ \text{SUBJ:} & \begin{bmatrix}\textit{np} \\ \text{ORTH: `Alice'}\end{bmatrix} \\ \text{OBJ:} & \boxed{1}\begin{bmatrix}\textit{np} \\ \text{ORTH: `Trudy'}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{QCAT:} & \text{WHO}\end{bmatrix} \tag{5}$$

An evaluation function is needed to determine which AVM is a better match for our query. We use WordNet [6] to identify the synonyms of ‘love’. The relevant part of the WordNet output is

```
Sense 2
love, enjoy
  ⇒ like
```

We construct a discrete pseudo-metric that returns 0 if for every mismatch, the words are synonym to each other, and 1 otherwise. We can now easily conclude that that AVM 4 is a closer match to AVM 1 than AVM 5 is because the distance between AVM 4 and AVM 1 is 0, while the distance between AVM 5 and AVM 1 is 1.

In this simplified example, we have shown how the combination of relaxed unification with HPSG can yield better results than using HPSG with classical unification. We have further shown how to construct a simple pseudo-metric based on WordNet.

3 System Architecture

A typical question-answering system consists of three major components:

1. an information retrieval (IR) search engine that locates relevant passages,
2. an answer extraction engine that extracts answers from the passages, and
3. an evaluation module that ranks the final answers and generates output according to specifications.

Our framework follows this high-level architecture. We further divide the second and the third components into two modules each. The resulting five modules are: IR search engine, HPSG parser, relaxed unification engine, ranking module, and output module. Figure 1 shows the data flow between these modules.

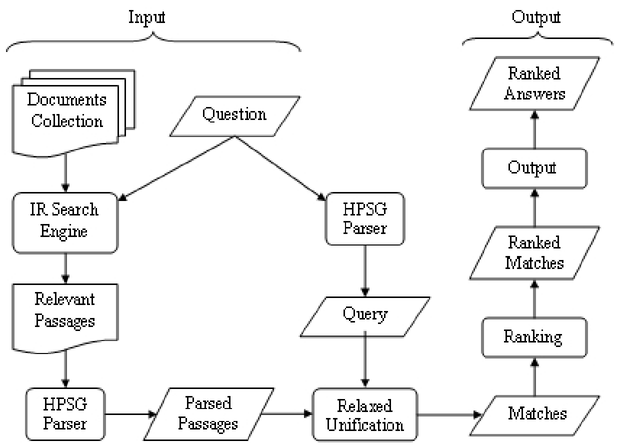

Figure 1: Question-Answering Framework Diagram.

The input to the system is a question and a collection of documents where an answer may be found. The input is presented to the IR search engine that searches the document collection and retrieves passages likely to contain an answer to the question according to a relevance measure. AnswerBus [13] and MultiText [5] are examples of IR search engines that can be used in this module. The question is parsed using an HPSG parser such as Stefy [7] to produce a query AVM. The retrieved passages are also parsed to create the corresponding AVMs. The passages AVMs are matched with the query in the relaxed unification module. This module incorporates relaxed metrics to determine the degree of matching. A threshold dictates the the point beyond which AVMs are deemed irrelevant and are discarded to reduce the search space. Several heuristic ranking criteria, such as structure and data metrics, are applied to the matches in the ranking module. The output module extracts answers from the ranked matches and formats them according to the output specifications.

The primary focus of this work is on the relaxed unification engine. The modularity of the framework affords employing the system in numerous applications. For instance, the system can be used in the TREC Question Answering Track by using MultiText for passage retrieval, and formatting the output to meet the TREC specifications. Similarly, the system can be used to improve performance of a web-based answering system such as AnswerBus by using AnswerBus for passage retrieval, eliminating unrelated passages using the relaxed unification engine, reranking the remaining passages and presenting them in a web page.

4 Evaluation Framework

Once a prototype is implemented based on the framework and the concepts developed in this work, it is crucial to subject the prototype to objective evaluation. The prototype will be tested in standardized environment accepted by industry and academia. Objective evaluation in standardized environment would enable us to assess the strengths and weaknesses of our approach, as well as compare it to other approaches and state-of-the-art systems.

There are currently three major international forums for evaluating question-answering systems:

1. Question-Answering Track of the Text REtrieval Conference (TREC) [12];
2. NTCIR Cross-Language Question Answering (CLQA) [4]; and
3. Cross-Language Evaluation Forum (CLEF) [3].

CLQA focuses on QA systems that operate with the Japanese and Korean languages, as well as English. CLEF focuses on European languages. The primary focus of CLQA and CLEF is on evaluating question-answering systems in multi-lingual environment. TREC, on the other hand, focuses exclusively on the English language. Our interests are presently directed at building a framework and an advanced prototype for question answering in English. Hence, TREC is the most suitable venue for evaluating our work.

TREC Question-Answering Track is built around open-domain closed-class questions. The questions and document collections are not domain specific. The type of questions asked is limited to definition, factoid, and list questions (closed-class). Factoid questions are typically easier to answer than other types of questions. List questions are basically factoid question where the answer contains several facts, rather than one. Definition questions are similar to list questions in that the answer contains a list of facts; the difference lies in the wording of the question, e.g., "what is platinum?" List and definition questions are harder than factoid questions because the system must determine the relevance or interestingness of a piece of information to decide whether to include it in the answer or not. Questions that require common knowledge or inferring from facts contained in the document collection are too difficult for current best systems. However, the difficulty of the track is increasing every year. When the question answering track started in TREC-8, 1999, the answer format was a 50 or 200 byte snippet, and a submission was judged correct if the target answer was contained in the snippet. In 2001, NIL responses where introduced—questions for which an answer could not be found in the documents. List questions were added in TREC 2001. A separate context task was introduced in TREC 2001. Questions were grouped in related sequences. This task was abandoned because the performance of systems depended on their ability to answer the questions independently and did not benefit from context. Starting from TREC 2002, submission were counted as correct only if they contained an exact answer, as opposed to a snippet. TREC 2003 witnessed increased interest in list and definition questions. TREC 2004 introduced the notion of a target: a target object or person are given followed by a set of questions related to the target. TREC 2004 also introduced the *other* question which implicitly asks "what other information can you tell me about the target?"

The correct answers to questions in TREC are provided by human assessors. It was found over the years that different assessors provide different answers. On one hand, this phenomena is a clear indication of user differences, and that these differences should be taken into consideration by the question answering systems. On the other hand, it makes comparing the performance of various systems difficult. Analysis of results have showed that the relative difference in performance between question-answering systems can vary significantly depending on the assessors. However, the ambiguity is mostly in comparing systems of similar performance. System that consistently perform better than others normally show similar trends under different assessors.

5 Conclusion and Future Work

We propose a framework for question answering that is based on using HPSG with relaxed unification. We give an example to illustrate how to use relaxed unification with HPSG to answer questions that are difficult for current systems. We propose a modular architecture for implementing a question-answering system based on HPSG and relaxed unification, and identify TREC Question Answering Track as an objective testbed for evaluating our approach and comparing its performance to other question answering systems in common environment defined by leading government, industry, and academia research groups.

In order to implement a question-answering system based on our proposed framework, a finer-grained evaluation function is required that provides a measure of the degree of mismatch in AVMs being unified. Further, efficient handling of cycles in the AVM graphs is an open problem. While our aim is devise a general theoretical solution, we will also investigate practical heuristic solutions.

Acknowledgement. This works is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Post Graduate Scholarship (PGS B).

References

[1] T. Abou-Assaleh. Theory of relaxed unification – proposal. Master's thesis, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, December 2002.

[2] T. Abou-Assaleh, N. Cercone, and V. Kešelj. An overview of the theory of relaxed unification. In *Proceedings of the International Conference on Advances in the Internet, Processing, Systems, Interdisciplinary Research, IPSI-2003*, Sveti Stefan, Montenegro, Former Yugoslavia, October 2003.

[3] Cross-Language Evaluation Forum (CLEF). WWW URL: `http://clef.isti.cnr.it`, March 2005.

[4] Cross Language Question Answering (CLQA). WWW URL: `http://www.slt.atr.jp/CLQA`, March 2005.

[5] G. V. Cormack, C. L. A. Clarke, C. R. Palmer, and R. C. Good. The MultiText retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, page 334, Berkeley, California, 1999. ACM Press.

[6] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[7] V. Kešelj. Stefy: Java parser for HPSGs, version 0.1. Technical Report CS-99-26, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, July 2000.

[8] V. Kešelj. Question answering using unification-based grammar. In *Advances in Artificial Intelligence, AI 2001*, Ottawa, Canada, June 2001. Springer.

[9] K. Knight. Unification: A multidisciplinary survey. *ACM Computing Surveys*, 21(1):93–124, 1989.

[10] C. J. Pollard and I. A. Sag. *Head-driven phrase structure grammar*. University of Chicago Press, Chicago, 1994.

[11] J. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[12] Text REtrieveal Conference (TREC). WWW URL: `http://trec.nist.gov`, March 2005.

[13] Z. Zheng. AnswerBus question answering system. In *Proceedings of the Human Language Technology Conference (HLT 2002)*, San Diego, CA, March 2002.