Question Generation with Minimal Recursion Semantics

Xuchen Yao† and Yi Zhang†‡

†Department of Computational Linguistics & Phonetics, Saarland University
‡LT-Lab, German Research Center for Artificial Intelligence
Saarbrücken, Germany
{xuchen,yzhang}@coli.uni-saarland.de

Abstract. This paper proposes a semantics-based approach to question generation by transforming the Minimal Recursion Semantics representation of declarative sentences to that of interrogative sentences. Sentences are first decomposed into smaller units on the semantic level. Then the semantic representation of target words are mapped to that of question words. Finally generation is realized through a linguistically deep English grammar. A prototype system is developed to verify the feasibility.

Key words: question generation, deep linguistic grammar, Minimal Recursion Semantics.

1 Introduction

Question Generation (QG) is the task of generating reasonable questions from a text. In terms of target complexity, the types of QG can be divided into *deep* QG (with deep questions, such as *why, what-if, how* questions) and *shallow* QG (with shallow questions, such as *who, what, when, where, which, how many/much, yes/no* questions) ([12]).

Different systems have been proposed or implemented to facilitate QG research and applications. These systems can be divided into mainly three categories: template-based ([9]), syntax-based ([14], [8]) and semantics-based ([13]). Template-based approaches are mostly suitable for applications with a special purpose, which sometimes comes within a closed-domain. The tradeoff between coverage and cost is hard to balance because human labors must be involved to produce high-quality templates. Syntax-based approaches are rather effective, especially for short sentences. The whole generation is based on tree nodes matching and operation. All operations are straight-forward from a syntactic point of view. However, the computer does this without knowing any underlying meaning of the transformed sentence. Also, it sometimes generates ungrammatical questions, which come from the surface realization of transformed trees thus do not always guarantee grammaticality.

While the first two kinds have already been applied, the third approach is theoretically more interesting and practically challenging. Following [13], we propose a semantics-based method of transforming the Minimal Recursion Semantics (MRS, [4]) representation of declarative sentences to that of interrogative sentences. As a subtask of semantics-based QG, the main efforts would be put to develop algorithms of decomposing complex semantic representations into smaller units and relations between them, which can be reused by other NLP tasks such as natural language understanding.

A set of tools from the DELPH-IN[1] community are assembled to help fulfill this purpose. Specifically, the MRS analysis is obtained from PET (a platform for experimentation with efficient HPSG processing techniques, [1]) while the generation function goes to the Linguistic Knowledge Builder (LKB, [5]). The underlying core linguistic component is the English Resource Grammar (ERG, [7]), a precision-oriented broad-coverage linguistic grammar in the framework of HPSG ([11]). Section 2 gives a more general introduction of these components while Section 3 specifies the system architecture. We address future work and conclude in Section 4.

2 Background

Minimal Recursion Semantics (MRS, [4]) is a meta-level language for describing semantic structures in some underlying object language. It can be implemented in typed feature structures and a bag of Elementary Predications (EPS) are its main components. An Elementary Predication (EP) is a single relation with its arguments, such as `chase(e,x,y)`, while the first argument (or ARG0), `e`, is called the bound variable. In the context of specific grammars that has an MRS representation, such as the ERG, the bound variable `e` in `chase(e,x,y)` denotes an event, following a form of Davidsonian representation in which all verbs introduce events. The capability of underspecifying out-scoping relations is an attractive feature of MRS and makes it a convenient semantic representation in large scale grammar engineering.

The English Resource Grammar (ERG, [7]) is a general-purpose broad-coverage grammar implementation under the HPSG framework. It consists of a large set of lexical entries under a hierarchy of lexical types, with a modest set of lexical rules for production. The Linguistic Knowledge Builder (LKB, [5]) is a grammar development environment for grammars in typed feature structures and unification-based formalisms. It can examine the competence and performance of a grammar by the means of parsing and generation. Central in our task, once given a valid MRS representation, linguistic realizations can be accomplished by chart generation ([3], [2]) in LKB.

Although ERG has a wide coverage of lexicons, there are always unknown words in real texts. Thus a high performance parsing system which incorporates statistical robust processing techniques is needed. We use PET ([1]), a platform for experimentation with efficient processing of unification-based grammars, to handle the parsing task efficiently and robustly. It employs a two-stage parsing model ([15]) with HPSG rules and PCFG models, balancing between precise linguistic interpretation and robust probabilistic coverage.

[1] Deep Linguistic Processing with HPSG: `http://www.delph-in.net/`

3 System Architecture

This section mainly describes the pipelines of the semantics-based question generator. It elaborates the core components: MRS decomposition for complex sentences and MRS transformation for simple sentences by examples. At last language independence and domain adaptability are addressed.

3.1 Overview

A prototype system based on semantics, MrsQG[2], is developed to constructs a general framework and test field for question generation on MRS, including modularized pre-processing, MRS manipulation, parsing and generation etc. Fig. 1 shows the processing pipeline. The following is a brief description of each step.

1. Term extraction. Terms are extracted as answer candidates. The Stanford Named Entity Recognizer ([6]), a regular expression NE tagger, an Ontology NE tagger are used to extract terms.
2. FSC construction. The Feature Structure Chart (FSC) format[3] is an XML-based format that introduces tokenization and external annotation to the ERG grammar and PET parser. Using FSC makes the terms annotated by NERs known to the parser. Thus all terms, no matter how long it is, are treated as an un-splittable token in the initial parsing chart.
3. Parsing with PET. PET accepts FSC and outputs MRS structures. Individual components are communicated through internal XML representations.
4. MRS decomposition. For complex sentences, it needs to be first broken into shorter ones with valid and meaningful semantic representation. Also, this shorter semantic representation must be able to generate outputs. Details in Section 3.2.
5. MRS transformation. With a valid MRS of a sentence, transformations are made to replace EPS for named entities with EPS for (WH) question words. Details in Section 3.3.
6. Generating with LKB. A two-phase generation algorithm ([3], [2]) is used.
7. Output selection. From a well-formed MRS, LKB might give multiple output. Some output might not sound fluent due to the fact that the ERG generates all linguistically plausible realizations. Thus various ranking guidelines are implemented to select the best one.
8. Output to console/XML. Depending on the purpose, MrsQG outputs to console for user interaction or XML files for formal evaluation.

The following sections describes more on step 4 and 5.

[2] http://code.google.com/p/mrsqg/

[3] http://wiki.delph-in.net/moin/PetInputFsc

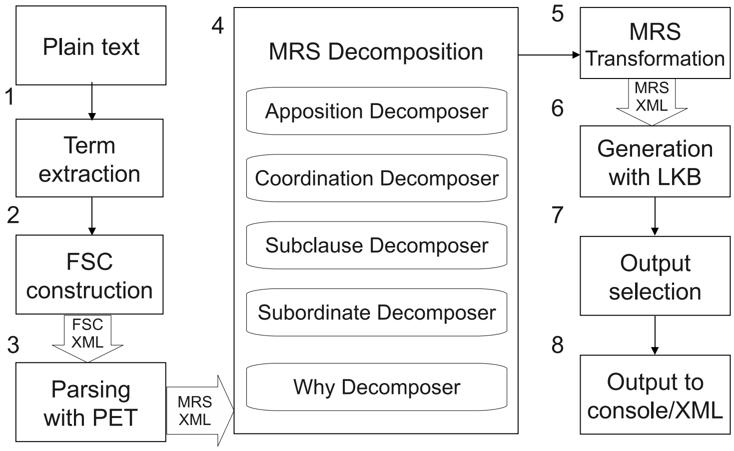
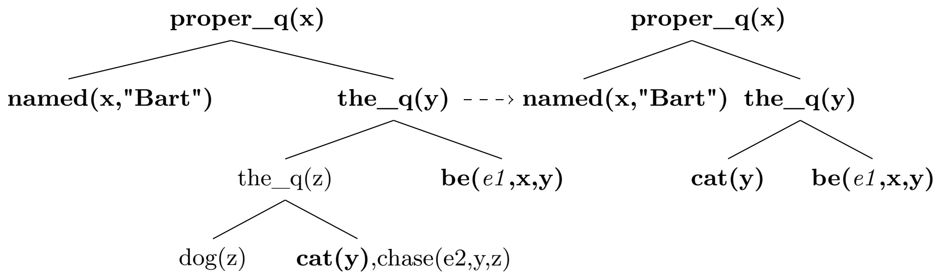
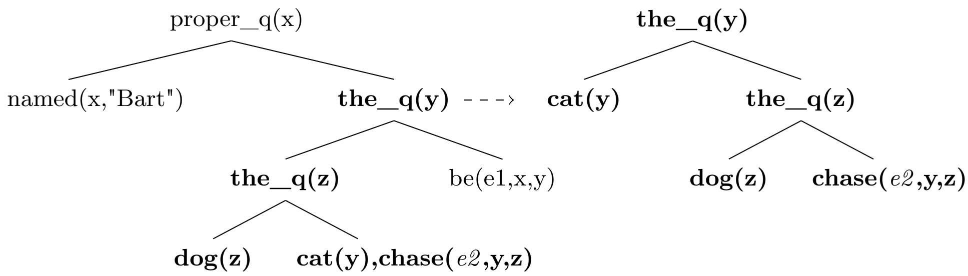


Fig. 1: The pipeline of a semantics-based question generation system.



(a) “Bart is the cat that chases the dog.” → “Bart is the cat.”



(b) “Bart is the cat that chases the dog.” → “The cat chases the dog.”

Fig. 2: A tree display of scoped MRS decomposition for subclauses. Upper-level tree nodes outscope lower-level nodes. The common suffix `_rel` is removed from all relations to save space. Scopes of each sentence is re-constructed after decomposition.

3.2 MRS Decomposition for Complex Sentences

A sentence that is too long generates lengthy questions, which is not desirable. Thus the semantic representations of complex sentences are first decomposed into partial and simpler ones. MrsQG employs four decomposers for apposition, coordination, subclause and subordinate clause. An extra WHY decomposer splits a causal sentence into two parts, reason and result, by extracting the arguments of the causal conjunction word, such as "because", "the reason", etc.

Fig. 2 shows an example of how the subclause decomposer works. Recall that the Davidsonian representation of MRS requires all verbs introduce events. Thus we assume every verb in a sentence represents an event, which is in the form of a sentence. The decomposer first identifies the event indicator (i.e. the verb[4]) in a sentence, such as "be" and "chase", then extract arguments of the verb and form new sentences for each verb. For instance, `be(e1,x,y)` takes two arguments, meaning "`x` is `y`". All EPs that take `x` and `y` as their bound variable are extracted and a new sentence "Bart is the cat" is assembled.

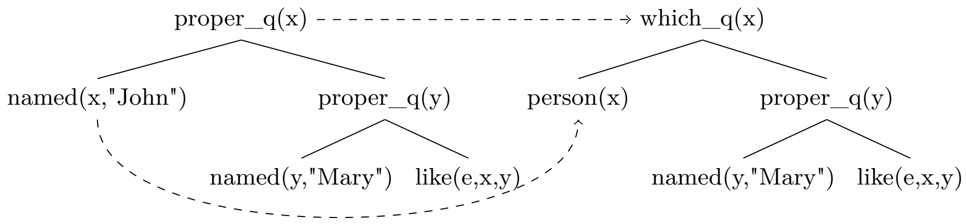
In the ontology of ERG, different linguistic structures are assign specific relations, such as `appos_rel` for apposition, `subord_rel` for subordinate clause. Thus very similar to the subclause decomposer, MRS decomposition spots these relations and extract their arguments to form new ones. In the current stage, whether the extracted semantic representation is true against the original one is not verified, which by itself is a research question for future investigation.

3.3 MRS Transformation for Simple Sentences

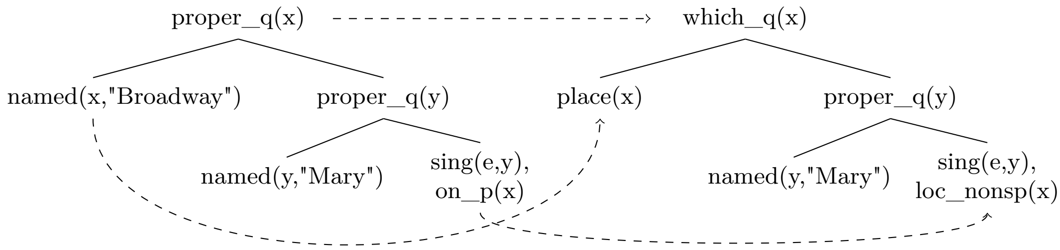
The transformation from declarative sentences into interrogatives follows a mapping between elementary predications (EPS) of relations. Fig. 3 has shown this mapping. Most terms in preprocessing are tagged as proper nouns (NNP or NNPS). Thus the EPS of a term turns out to consist of two EPs: `proper_q_rel` (a quantification relation) and `named_rel` (a naming relation), both of which have the same bound variable while `proper_q_rel` outscopes `named_rel`. The EPS of WH-question words have a similar parallel. For instance, the EPS of "who" consists of two relations: `which_q_rel` and `person_rel`, both of which also have the same bound variable while `which_q_rel` outscopes `person_rel`. Changing the EPS of terms to EPS of WH-question words naturally results in an MRS for WH-questions.

Similarly, in WHERE/WHEN/WHY questions, the EPS for the question word are `which_q_rel` and `place_rel/time_rel/reason_rel`. Special attentions must be paid to the preposition word that usually comes before location/time. A preposition word stays on the same node of a semantic tree as the head of the phrase this PP is attached to (as shown in Fig. 3(bcd)). The EP of the preposition must be changed to a `loc_nonsp_rel` EP (an implicit locative which does not specify a preposition) which takes the WH word relation as an argument in both cases of WHEN/WHERE. This EP avoids generating non-grammatical question words such as "in where" and "on when".

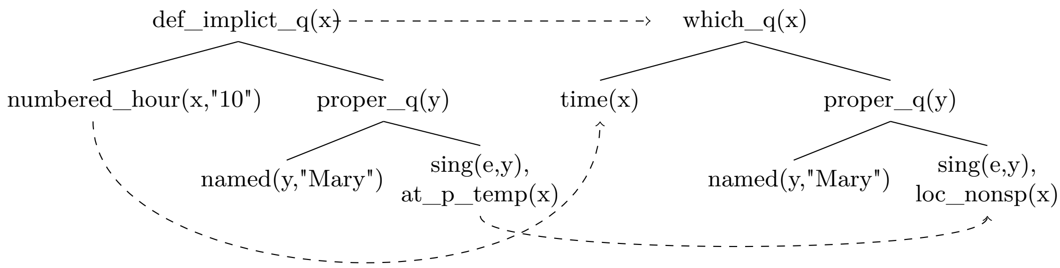
[4] The actual implementation is more complicated because there are other words such as non-scopal adverbs that also take events as bound variables.



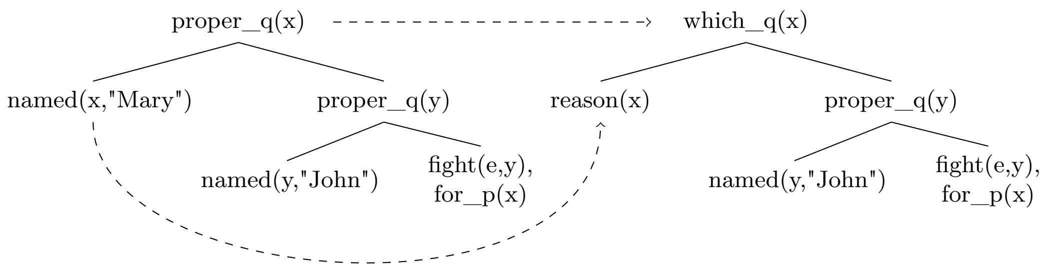
(a) “John likes Mary” → “Who likes Mary?”



(b) “Mary sings on Broadway.” → “Where does Mary sing?”



(c) “Mary sings at 10.” → “When does Mary sing?”



(d) “John fights for Mary.” → “Why does John fight?”

Fig. 3: A tree display of scoped MRS relations and corresponding interrogative forms. Upper-level tree nodes outscope lower-level nodes. Some EPs are simplified to save space.

3.4 Language Independence and Domain Adaptability

MrsQG aims to stay language-neutral based on a semantics transformation of sentences. In principle, it needs little modification to adapt to other languages, as long as there is a grammar[5] conforming with the HPSG structure and LKB. However, the experience on multi-lingual grammar engineering has shown that although MRS offers a higher level of abstraction from the syntax, it is difficult to guarantee absolute language independence. As a syntax-semantics interface, part of the MRS representation will inevitably carry some language specificity. As a consequence, the MRS transfer needs to be adapted for the specific grammars, similar to the situations in MRS-based machine translation ([10]).

The domain adaptability is confined in the following parts:

1. Named entity recognizers (NER). For a different domain, the Stanford NER must be re-trained. MrsQG also uses an ontology NER. Thus collections of domain-specific named entities can be easily plugged-in to MrsQG.
2. HPSG parser. The PET parser needs to be re-trained on a new domain with an HPSG treebank. However, since the underlying HPSG grammars are mainly hand-written, they normally generalize well and have a steady performance on different domains.

4 Conclusion

We report on MrsQG, a semantics-based question generation prototype system participated in the the Question Generation Shared Task Evaluation Challenge 2010. The core technology of the system is built upon the idea of MRS-transfer. The system involves heavy machinery, including various preprocessing, parsing and generation with a linguistically deep grammar, and MRS rewriting. To our knowledge, this is the first implementation of a working system for the question generation task following the idea of [13]. The system is theoretically interesting in that it involves a chain of deep processing steps. With the help of a precision grammar (ERG), the grammaticality of the generation outputs is guaranteed. The manipulation on the semantic structure also allows one to produce various meaningful questions.

The development of the system also shows that there are various challenging research and engineering questions. Particularly, the MRS transfer component turns out to be fragile (i.e. a slightly ill-formed MRS will lead to generation failure). Also, the ranking of the generation outputs can be further improved by incorporating language models trained on large corpora. The multi-linguality and cross-domain applicability of the approach needs to be investigated in future research.

Acknowledgement We thank for discussions and help from Dr. Dan Flickinger and the DELPH-IN community. The first author acknowledges financial support

[5] For a list of available grammars, check `http://wiki.delph-in.net/moin/MatrixTop`

from the Erasmus Mundus Program through scholarships for the European Masters Program in Language and Communication Technologies (LCT). The second author thanks the German Excellence Cluster of Multimodal Computing and Interaction for the support of the work.

References

1. Ulrich Callmeier. PET – a platform for experimentation with efficient HPSG processing techniques. *Nat. Lang. Eng.*, 6(1), 2000.
2. J. Carroll and S. Oepen. High efficiency realization for a wide-coverage unification grammar. *Lecture notes in computer science*, 3651:165, 2005.
3. John Carroll, Ann Copestake, Dan Flickinger, and Victor Poznanski. An efficient chart generator for (semi-)lexicalist grammars. In *Proceedings of the 7th European Workshop on Natural Language Generation (EWNLG'99)*, Toulouse, France, 1999.
4. A. Copestake, D. Flickinger, C. Pollard, and I.A. Sag. Minimal recursion semantics: An introduction. *Research on Language & Computation*, 3(4):281–332, 2005.
5. Ann Copestake. *Implementing Typed Feature Structure Grammars*. CSLI: Stanford, 2002.
6. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA, 2005.
7. Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28, 2000.
8. M. Heilman and N. A. Smith. Question Generation via Overgenerating Transformations and Ranking. Technical report, Language Technologies Institute, Carnegie Mellon University Technical Report CMU-LTI-09-013, 2009.
9. Jack Mostow and Wei Chen. Generating instruction automatically for the reading strategy of self-questioning. In *Proceeding of the 2009 conference on Artificial Intelligence in Education*, Amsterdam, The Netherlands, 2009.
10. Stephan Oepen, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, and Victoria Rosén. Som å kapp-ete med trollet? Towards MRS-based Norwegian-English Machine Translation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, Baltimore, MD, October 2004.
11. C.J. Pollard and I.A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, 1994.
12. V. Rus and A. Graesser, editors. *The Question Generation Shared Task and Evaluation Challenge*. 2009.
13. Ivan A. Sag and Dan Flickinger. Generating questions with deep reversible grammars. In *Proceedings of the First Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA: NSF, 2008.
14. Brendan Wyse and Paul Piwek. Generating Questions from OpenLearn study units. In *Proceedings of the 2nd Workshop on Question Generation In Craig, S.D. & Dicheva, S. (Eds.) (2009) AIED 2009: 14th International Conference on Artificial Intelligence in Education: Workshops Proceedings*, 2009.
15. Yi Zhang, Stephan Oepen, and John Carroll. Efficiency in unification-based N-best parsing. In *IWPT '07: Proceedings of the 10th International Conference on Parsing Technologies*, pages 48–59, Morristown, NJ, USA, 2007.