LIMSIILES: Basic English Substitution for Student Answer Assessment at SemEval 2013

Martin Gleize
LIMSI-CNRS & ENS
B.P. 133 91403 ORSAY CEDEX, France
gleize@limsi.fr

Brigitte Grau
LIMSI-CNRS & ENSIIE
B.P. 133 91403 ORSAY CEDEX, France
bg@limsi.fr

Abstract

In this paper, we describe a method for assessing student answers, modeled as a paraphrase identification problem, based on substitution by Basic English variants. Basic English paraphrases are acquired from the Simple English Wiktionary. Substitutions are applied both on reference answers and student answers in order to reduce the diversity of their vocabulary and map them to a common vocabulary. The evaluation of our approach on the SemEval 2013 Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge data shows promising results, and this work is a first step toward an open-domain system able to exhibit deep text understanding capabilities.

1 Introduction

Automatically assessing student answers is a challenging natural language processing task (NLP). It is a way to make test grading easier and improve adaptive tutoring (Dzikovska et al., 2010), and is the goal of the SemEval 2013's task 7, titled *Joint Student Response Analysis*. More specifically, given a question, a known correct "reference answer" and a 1- or 2-sentence student answer, the goal is to determine the student's answer accuracy (Dzikovska et al., 2013). This can be seen as a paraphrase identification problem between student answers and reference answers.

Paraphrase identification searches whether two sentences have essentially the same meaning (Culicover, 1968). Automatically generating or extracting semantic equivalences for the various units of language – words, phrases, and sentences – is an important problem in NLP and is being increasingly employed to improve the performance of several NLP applications (Madnani and Dorr, 2010), like question-answering and machine translation.

Paraphrase identification would benefit from a precise and broad-coverage semantic language model. This is unfortunately difficult to obtain to its full extent for any natural language, due to the size of a typical lexicon and the complexity of grammatical constructions. Our hypothesis is that the simpler the language lexicon is, the easier it will be to access and compare meaning of sentences. This assumption is justified by the multiple attempts at controlled natural languages (Schwitter, 2010) and especially simplified forms of English. One of them, Basic English (Ogden, 1930), has been adopted by the Wikipedia Project as the preferred language of the Simple English Wikipedia[1] and its sister project the Simple English Wiktionary[2].

Our method starts with acquiring paraphrases from the Simple English Wiktionary's definitions. Using those, we generate variants of both sentences whose meanings are to be compared. Finally, we compute traditional lexical and semantic similarity measures on those two sets of variants to produce features to train a classifier on the SemEval 2013 datasets in order to take the final decision.

2 Acquiring simplifying paraphrases

Simple Wiktionary word definitions are different from usual dictionary definitions. Aside from the

[1] http://simple.wikipedia.org
[2] http://simple.wiktionary.org

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 598–602, Atlanta, Georgia, June 14-15, 2013.

simplified language, they often prefer to give a complete sentence where the word – e.g. a verb – is used in context, along with an explanation of what it means. To define the verb *link*, Simple Wiktionary states that *If you link two or more things, you make a connection between them* (1), whereas the standard Wiktionary uses the shorter and more cryptic *To connect two or more things*.

We notice in this example that the definition from Simple Wiktionary consists of two clauses, linked by a subordination relation. It's actually the case for a lot of verb definitions: a quick statistical study shows that 70% of these definitions are composed of two clauses, an independent clause, and a subordinate clause (often an adverbial clause). One clause illustrates how the verb is used, the other gives the explanation and the actual dictionary definition, as in example (1). These definitions are the basis of our method for acquiring paraphrases.

2.1 Pre-processing

We use the Stanford Parser to parse the definitions and get a dependency graph (De Marneffe and Manning, 2008). Using a few hand-written rules, we then retrieve both parts of the definition, which we call the *word part* and the *defining part* (see table 1 page 3 for examples). We can do this for definitions of verbs, but also for nouns, like *the giraffe is the tallest land animal in the world* to define *giraffe*, or adjectives, like *if something is bright it gives out or fills with much light* to define *bright*. We only provide the details of our method for processing verb definitions, as they correspond to the most complex cases, but we proceed similarly for noun, adjective and adverb definitions.

2.2 Argument matching

Word and defining parts alone are not paraphrases, but we can obtain phrasal paraphrases from them. If we see word part and defining part as two semantically equivalent predications, we have to identify the two predicates with their arguments, then match arguments with corresponding meaning, i.e. match arguments which designate the same entity or assume the same semantic function in both parts, as showed in Table 2.

For verb definitions, we identify the predicates as the main verbs in both clauses (hence *link* matching with *make* in table 2) and their arguments as a POS-filtered list of their syntactic descendants. Then, our assumption is that every argument of the word part predicate is present in the defining part, and the defining part predicate can have extra arguments (like *a connection*).

you	→	you
link	→	make
∅	→	a connection
∅	→	between
two or more things	→	them

Table 2: Complete matching for the definition of verb *link*

We define $s(A, B)$, the *score* of the pair of arguments (A, B), with argument A in the word part and argument B in the defining part. We then define a *matching* M as a set of such pairs, such that every element of every possible pair of arguments is found at most one time in M. A *complete matching* is a matching M that matches every argument in the word part, i.e., for each word part argument A, there exists a pair of arguments in M which contains A. Finally, we compute the *matching score* of M, $S(M)$, as the sum of scores of all pairs of M.

The *score* function $s(A, B)$ is a hand-crafted linear combination of several features computed on a pair of arguments (A, B) including:

- Raw string similarity. Sometimes the same word is reused in the defining part.
- Having an equal/compatible dependency relation with their respective main verb.
- Relative position in clause.
- Relative depth in parsing tree. These last 3 features assess if the two arguments play the same syntactic role.
- Same gender and number. If different, it's unlikely that the two arguments designate the same entity.
- If (A, B) is a pair (noun phrase, pronoun). We hope to capture an anaphoric expression and its antecedent.

Word (POS-tag)	Word part	Defining part
link (V)	you link two or more things	you make a connection between them
giraffe (N)	the giraffe	the tallest land animal in the world
bright (Adj)	something is bright	it gives out or fills with much light

Table 1: Word part and defining part of some Simple Wiktionary definitions

- WordNet similarity (Pedersen et al., 2004). If words belong to close synsets, they're more likely to identify the same entity.

2.3 Phrasal paraphrases

We compute the complete matching M which maximizes the matching score $S(M)$. Although it is possible to enumerate all matchings, it is intractable; therefore when predicates have more than 4 arguments, we prefer constructing a best matching with a beam search algorithm. After replacing each pair of arguments with linked variables, and attaching unmatched arguments to the predicates, we finally obtain phrasal paraphrases of this form:

$\langle\ X$ link Y, X make a connection between $Y\ \rangle$

3 Paraphrasing exercise answers

3.1 Paraphrase generation and pre-ranking

Given a sentence, and our Simple Wiktionary paraphrases (about 20,650 extracted paraphrases), we can generate sentential paraphrases by simple syntactic pattern matching –and do so recursively by taking previous outputs as input–, with the intent that these new sentences use increasingly more Basic English. We generate as many variants starting from both reference answers and student answers as we can in a fixed amount of time, as an anytime algorithm would do. We prioritize substituting verbs and adjectives over nouns, and non Basic English words over Basic English words.

Given a student answer and reference answers, we then use a simple Jaccard distance (on lowercased lemmatized non-stopwords) to score the closeness of student answer variants to reference answer variants: we measure how close the vocabulary used in the two statements has become. More specifically, for each reference answer A, we compute the n closest variants of the student answer to A's variant set. In our experiments, $n = 10$. We finally rank the reference answers according to the average distance from their n closest variants to A's variant set and keep the top-ranked one for our classification experiment. Figure 1 illustrates the whole process.

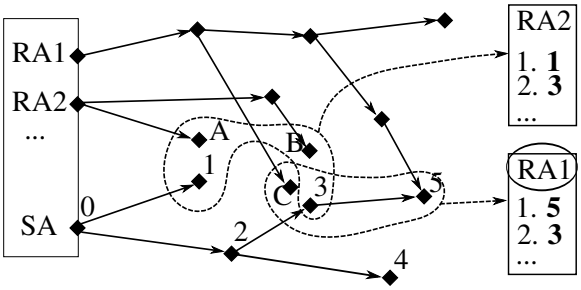

Figure 1: Variants are generated from all reference answers (RA) and the student answer (SA). For each reference answer RA, student answer variants are ranked based on their lexical distance from the variants of RA. The reference with the n closer variants to the student variants is kept (here: RA1).

3.2 Classifying student answers

SemEval 2013 task 7 offers 3 problems: a 5-way task, with 5 different answer judgements, and 3-way and 2-way tasks, conflating more judgement categories each time. Two different corpora, Beetle and SciEntsBank, were labeled with the 5 following labels: Correct, Partially_correct_incomplete, Contradictory, Irrelevant and Non_Domain, as described in (Dzikovska et al., 2012). We see the n-way task as a n-way classification problem. The instances of this problem are the pairs (student answer, reference answer).

We compute for each instance the following features: For each of the n closest variants of the student answer to some variant of the reference answer computed in the pre-ranking phase:

- Jaccard similarity coefficient on non-stopwords.
- A boolean representing if the two statements have the same polarity or not, where polarity

is defined as the number of *neg* dependencies in the Stanford Parser dependency graph.

- Number of "paraphrasing steps" necessary to obtain the variant from a raw student answer.
- Highest WordNet similarity of their respective nouns.
- WordNet similarity of the main verbs.

General features:

- Answer count (how many students typed this answer), provided in the datasets.
- Length ratio between the student answer and the closest reference answer.
- Number of (non-stop)words which appear neither in the question nor the reference answers.

We train an SVM classifier (with a one-against-one approach to multiclass classification) on both Beetle and SciEntsBank, for each n-way task.

3.3 Evaluation

Table 3 presents our system's overall accuracy on the 5-way task, along with the top scores at SemEval 2013, mean scores, and baselines –majority class and lexical overlap– described in (Dzikovska et al., 2012).

System	Beetle unseen answers	SciEntsBank unseen questions
Majority	0.4010	0.4110
Lexical overlap	0.5190	0.4130
Mean	0.5326	0.4078
ETS-run-1	0.5740	**0.5320**
ETS-run-2	**0.7150**	0.4010
Simple Wiktio	0.5330	**0.4820**

Table 3: SemEval 2013 evaluation results.

Our system performs slightly better in overall accuracy on Beetle unseen answers and SciEntsBank unseen questions than both baselines and the mean scores. While results are clearly below the best system trained on the Beetle corpus questions, we hold the third best score for the 5-way task on SciEntsBank unseen questions, while not fine-tuning our system specifically for this corpus. This is rather encouraging as to how suitable Simple Wiktionary is as a resource to extract open-domain knowledge from.

4 Discussion

The system we present in this paper is the first step towards an open-domain machine reading system capable of understanding and reasoning. Direct modeling of the semantics of a full natural language appears too difficult. We therefore decide to first project the English language onto a simpler English, so that it is easier to model and draw inferences from.

One complementary approach to a minimalistic language model, is to accept that texts are replete with gaps: missing information that cannot be inferred by reasoning on the text alone, but require a certain amount of background knowledge. Penas and Hovy (2010) show that these gaps can be filled by maintaining a background knowledge base built from a large corpus.

Although Simple Wiktionary is not a large corpus by any means, it can serve our purpose of acquiring basic knowledge for assessing exercise answers, and has the advantage to be in constant evolution and expansion, as well as interfacing very easily with the richer Wiktionary and Wikipedia.

Our future work will be focused on enriching and improving the robustness of our knowledge acquisition step from Simple Wiktionary, as well as introducing a true normalization of English to Basic English.

Acknowledgments

We acknowledge the Wikimedia Foundation for their willingness to provide easily usable versions of their online collaborative resources.

References

P.W. Culicover. 1968. *Paraphrase generation and information retrieval from stored text*. In Mechanical Translation and Computational Linguistics, 11(12), 7888.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. *Stanford typed dependencies manual*. Technical report, Stanford University.

Myroslava O. Dzikovska, Diana Bental, Johanna D. Moore, Natalie Steinhauser, Gwendolyn Campbell, Elaine Farrow, and Charles B. Callaway. 2010. *Intelligent tutoring with natural language support in the BEETLE II system*. In Proceedings of Fifth European Conference on Technology Enhanced Learning (EC-TEL 2010), Barcelona.

Myroslava O. Dzikovska, Rodney D. Nielsen and Chris Brew. 2012. *Towards Effective Tutorial Feedback for Explanation Questions: A Dataset and Baselines*. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2012), Montreal.

Myroslava O. Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan and Hoa Trang Dang. 2013. *SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge*. In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013), in conjunction with the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013). Atlanta, Georgia, USA. 13-14 June.

Nitin Madnani and Bonnie J. Dorr. 2010. *Generating phrasal and sentential paraphrases: A survey of data-driven methods*. In Computational Linguistics 36 (3), 341-387.

Charles Kay Ogden. 1930. *Basic English: A General Introduction with Rules and Grammar*. Paul Treber, London.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. *WordNet::similarity–measuring the relatedness of concepts*. In Proceedings of the Nineteenth National Conference on Artificial Intelligence(AAAI-04), pages 10241025.

Anselmo Penas and Eduard H. Hovy. 2010. *Filling Knowledge Gaps in Text for Machine Reading*. COLING (Posters) 2010: 979-987, Beijing.

Rolf Schwitter. 2010. *Controlled Natural Languages for Knowledge Representation*. COLING (Posters) 2010, Beijing.