IUCL: Combining Several Information Sources for SemEval Task 5

Alex Rudnick, Levi King, Can Liu, Markus Dickinson, Sandra Kübler
Indiana University
Bloomington, IN, USA
{alexr,leviking,liucan,md7,skuebler}@indiana.edu

Abstract

We describe the Indiana University system for SemEval Task 5. The system is based on combining different information sources to arrive at a final L2 guess, incorporating: phrase tables, an L2 language model, and collocational tendencies. We also consider different data sources.

1 Introduction

The task of translating an L1 fragment occurring in the midst of an L2 sentence is one in which a phrase occurs in an already-rich target language (L2) context; this makes the task quite different from translation in the general case. For this reason, we decided to approach the problem by combining standard Machine Translation technology with target language information, such as contextual relationships. This is broken down into various steps: 1) constructing candidate translations for the L1 fragment, including weights for the likelihood of each translation; 2) scoring candidate translations via a language model of the L2; 3) scoring candidate translations via dependency-driven word similarity measure (Lin, 1998) (in this paper we call it SIM) and 4) combining the scores from 1)-3) via minimized error rate training (MERT), to arrive at a final solution. Step 1) models transfer knowledge between the L1 and L2; step 2) models facts about the L2 grammar, i.e., what translations fit well into the local context; step 3) models collocational and semantic tendencies of the L2; and step 4) gives different weights to each of the three sources of information. Although we did not finish step 3) in time for the official results, we report it here, as it represents the most novel aspect of the system – namely, the exploitation of the rich L2 context – and it results in our team's best system. In general, our system is fully language-independent, with accuracy varying due to the size of data sources and quality of input technology (e.g., syntactic parse accuracy).

2 Data Sources

The data sources serves two major steps of our system: 1) For L2 candidate generation we used Europarl, Bebelnet 2) For candidate ranking using L1 context we used Wikipedia, Google Books Syntactic N-grams. The data sources are described below:

Europarl The Europarl Parallel Corpus is a corpus of proceedings of the European Parliament. It is a good resource for constructing translation phrase tables because the corpus contains 21 European Languages, and it is aligned sentence by sentence. From this corpus, we built phrase tables for English-Spanish, English-German, French-English, Dutch-English.

Babelnet In the case our constructed phrase-tables do not contain a translation for a source phrase, we need to split up the source phrase into smaller components and look up each components from Babelnet.
- we used: University of Pisa wikipedia extractor - MD: make clear which data sources were for which parts (quick point-aheads should be good)

Wikipedia For German and Spanish, we used recent Wikipedia dumps. To save time during parsing, sentences longer than 25 words were removed. The remaining sentences were POS tagged and dependency parsed using Mate Parser and pre-trained models (Bohnet, 2010; Bohnet and Kuhn, 2012; Seeker and Kuhn, 2013). To keep our English Wikipedia data set comparable

LK: Alex may have the exact info for the dumps if needed in size to the German and Spanish sets, we chose an older (2006), smaller dump. Long sentences were removed, and the remaining sentences were POS tagged and dependency parsed using the pretrained Stanford Parser (Klein and Manning, 2003; de Marneffe et al., 2006). The resulting sizes of the data sets are (roughly): German: 389M words, 28M sentences; Spanish: 147M words, 12M sentences; English: 253M words, 15M sentences. Dependencies extracted from these parsed data sets served as training for the SIM system described in section 3.3.

(Goldberg and Orwant, 2013) says this contains 919M "items"; = dependency types? Taken from 3,473,595 books.

Google Books Syntactic N-grams For English, we also trained a SIM system on the arcs dataset of the Google Books Syntactic N-Grams (Goldberg and Orwant, 2013).

3 Our System

3.1 Constructing Candidate Translations

- GIZA++ and Moses for phrase table extraction

Given that this is essentially a local translation problem, we use as our starting point a phrase table constructed from parallel text, weighted with probabilities. We use GIZA++ (Och and Ney, 2000) and Moses (Koehn et al., 2007) to construct the phrase tables. The quality of the phrase table depends upon the size of the data–an issue we discuss with larger phrase tables in section XX–but in the case of missing phrases from the table, we back off to subphrases in the following way: ...

MD: 1. What is the exact back-off procedure? 2. Do we back off only in the case of missing phrases, or do we consider multiple possibilities for one phrase, even if its already in the table?

MD: we should give names to each model, to easily refer to them throughout the paper and in tables (IUCL1 and IUCL2 are the official submissions, but for other models we may want more descriptive names) alexr: the models are basically the same – we just changed the phrase tables for the English/German setup on the second run. I'll have to check the differences exactly, but there was some problem with the phrase tables for the first run – maybe we didn't run the whole Moses pipeline appropriately and tokenization/truecasing was messed up. But from an algorithmic perspective, they're the same.

MD: Okay, that makes sense - do we just concatenate the data sources to derive a phrase table for each language pair? Or is it just Europarl for this phase?

3.2 Scoring Candidate Translations via a L2 Language Model

To examine how well a phrase fit into a L1 context, the most intuitive method is to use a N-gram model built from the L1 and rank the candidate phrases using N-gram scores. With a large vocabulary, the construction and query of a N-gram language model could be very time consuming, we used KenLM Language Model Toolkit for efficiency (Heafield, 2011).

3.3 Scoring Candidate Translations via Dependency-Based Word Similarity

Definition In addition to a N-gram language model ranking, we could rank the candidate phrases based on how well each of the component fits into the L1 context. The fitness is measured in terms of dependency-based word similarity. We slightly adapted the word similarity measure in (Lin, 1998):

$$sim(w_1, w_2) = \frac{2\prod count(head,dep,label)}{count(head,-,label)+count(-,dep,label)}$$

Where $count(head, dep, label)$ is the frequency that a particular dependency triple occurs in the L1 corpus. $count(head, -, label)$ is the frequency that a word occurs as a head in relationship "label" with any dependent. $count(-, dep, label)$ is the frequency that a word occurs as a dependent in relationship "label" with any head.

The fitness of a phrase is the average word similarity over all its components. For example the fitness of phrase "eat with chopsticks" would be computed as:

$$\frac{fit(eac)+fit(with)+fit(chopsticks)}{3}$$

Depending on where the head/dependent of each phrase component lies, it could be outside the phrase or inside the phrase. Both cases are included in our calculation. This method focuses the calculation on tightly associated words by using only one words's head and dependent in the computation.

During training time, we obtained dependency-based word similarity statistics for all target languages using corresponding dependency triples. During testing time, the word similarity value is queried and combined with N-gram language model to rank the candidate phrase.

Storing and Caching The large vocabulary and huge number of combinations of "head, dependent, relationship" label poses an efficiency problem when querying the dependency-based word similarity values. Thus we stored the dependency triples in a database with a Python programming interface (sqlite3), and built database indexes on the frequent query types. However, for frequently searched dependency triples, re-querying the database is still inefficient. Thus we built a query cache to store the recently queries triples. Using the database and cache significantly speeds up our system.

Back-Off Lexical-based dependency suffers from data sparsity, so in addition to computing the lexical fitness of a phrase, we also calculate the POS fitness. For example POS fitness of "eat with chopsticks" would be computed as $\frac{fit(VBG)+fit(IN)fit(NNS)}{3}$

3.4 Tuning Weights with MERT

- ZMERT (Zaidan, 2009) - exact set-up

4 Experiments

4.1 Official System

(Bird et al., 2009)

4.2 Expansion #1: Experiments with PMI over Dependencies

4.3 Expansion #2: Experiments with Large Phrase Tables

- EU bookshop corpus - MultiUN corpus - you can find so many corpora on Opus!! (Tiedemann, 2012)

5 Conclusion

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Bernd Bohnet and Jonas Kuhn. 2012. The best of both worlds – A graph-based completion model for transition-based parsers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 77–87, Avignon, France.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 89–97, Beijing, China.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, Genoa, Italy.

Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of English books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 241–247, Atlanta, GA.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom, July.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL-2003*, pages 423–430, Sapporo, Japan.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *International Conference on Machine Learning (ICML)*, volume 98, pages 296–304.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 440–447, Hong Kong.

Wolfgang Seeker and Jonas Kuhn. 2013. Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39(1):23–55.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 2214–2218, Istanbul, Turkey.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.