SFS-TUE: Compound Paraphrasing with a Language Model and Discriminative Reranking

Yannick Versley
SfS / SFB 833
University of Tübingen
versley@sfs.uni-tuebingen.de

Abstract

This paper presents an approach for generating free paraphrases of compounds (task 4 at SemEval 2013) by decomposing the training data into a collection of templates and fillers and recombining/scoring these based on a generative language model and discriminative MaxEnt reranking.

The system described in this paper achieved the highest score (with a very small margin) in the (default) *isomorphic* setting of the scorer, for which it was optimized, at a disadvantage to the *non-isomorphic* score.

1 Introduction

Compounds are an interesting phenomenon in natural language semantics as they normally realize a semantic relation (between head and modifier noun) that is both highly ambiguous as to the type of relation and usually nonambiguous as to the concepts it relates (namely, those of the two nouns).

Besides inventory-based approaches, where the relation is classified into a fixed number of relations, many researchers have argued that the full variability of the semantic relations inherent in compounds is best captured with paraphrases: Lauer (1995) proposes to use a preposition as a proxy for the meaning of a compound. Finin (1980) and later Nakov (2008) and others propose less restrictive schemes based on paraphrasing verbs.

A previous SemEval task (task 9 in 2010; Butnariu et al., 2009). The most successsful approaches for this task such as Nulty and Costello (2010), Li et al. (2010), and Wubben (2010), or the subsequent approach of Wijaya and Gianfortoni (2011), all make efficient use of both the training data and general evidence from WordNet or statistics derived from large corpora. The paper of Li et al. mentions that solely inducing a global ranking of paraphrasing verbs from the training data (looking which verb is ranked higher in those cases where both were considered for the same compound) yielded higher scores than an unsupervised approach based on the semantic resources, underlining the need to combine training data and resources efficiently.

SemEval 2013 task 4 The present task on providing free paraphrases for noun compounds (Hendrickx et al., 2013) uses a dataset collected from Mechanical Turk workers asked to paraphrase a given compound (without context). Prepositional, verbal, and other paraphrases all occur in the data:

(1)
a. *bar* for *wine*
b. *bar* that serves *wine*
c. *bar* where *wine* is sold
d. sweet *vinegar* made from *wine*

In the examples, the words of the compound (*wine bar* and *wine vinegar*, respectively) are put in italics, and other content words in the paraphrase are underlined.

It is clear that certain paraphrases (*X* for *Y*) will be common across many compounds, whereas the ones containing more lexical material will differ even between relatively similar compounds (consider *wine bar* from the example, and *liquor store*, which allows paraphrase c, but not paraphrase b).

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 148–152, Atlanta, Georgia, June 14-15, 2013.

2 General Approach

The approach chosen in the SFS-TUE system is based on first retrieving a number of similar compounds, then extracting a set of building blocks (patterns and fillers) from these compounds, recombining these building blocks, and finally ranking the list of potential paraphrases. The final list is post-processed by keeping only one variant of each set of paraphrases that only differ in a determiner (e.g., '*strike* from *air*' and '*strike* from the *air*') in order to make a 1:1 mapping between system response and gold standard possible.

As a first step, the system retrieves the most similar compounds from the training data.

This is achieved Lin's wordnet similarity measure (Lin, 1998) using the implementation in NLTK (Bird et al., 2009). The similarity of two compounds X_1Y_1 and X_2Y_2 is calculated as

$$s_C = \min(sim(X_1, X_2), sim(Y_1, Y_2)) + 0.1 \cdot (sim(X_1, X_2) + sim(Y_1, Y_2))$$

which represents a compromise between requiring that both modifier and head are approximately similar, and still giving a small boost to pairs that have very high modifier similarity but low head similarity, or vice versa. For training, the target compound is excluded from the most-similar compounds list so that candidate construction is only based on actual neighbours.

The paraphrases for the most similar compound entries (such as 2a) are broken down into templates (2b) and fillers (2c), by replacing modifier and head by X and Y, respectively, and other content words by their part-of-speech tag.

(2) a. *bar* that serves *wine*
b. X that VBZ Y
c. VBZ:serve

Conversely, template fillers consist of all the extracted content words, categorized by their part-of-speech. (Part-of-speech tags were assigned using the Stanford POS tagger: Toutanova et al., 2003).

Both paraphrase templates and template fillers are weighted by the product of the similarity value s_C between the target compound and the neighbour, and the total frequency of occurrence in that neighbour's paraphrases. (For example, if Mechanical Turk participants named "*bar that sells wine*" twice and "*bar that serves wine*" once, the total frequency of "X that VBZ Y" would be three).

type	examples
Y_of	*Y* of *X* (159) / *Y* of the *X* (59) / *Y* of a *X* (47)
Y_for	*Y* for *X* (114) / *Y* for the *X* (33)
Y_VBZ	*Y* that VBZ *X* (91)/ *Y* which VBZ *X* (45)
Y_VBG	*Y* VBG *X* (90) / *Y* VBG the *X*/ *Y* VBG with *X*
Y_VBN	*Y* VBN for *X* (82) / *Y* VBN by *X* (52)
Y_in	*Y* in *X* (31)
Y_on	*Y* on *X* (38)

Table 1: Most frequent paraphrase pattern types and pattern instances

Paraphrase candidates are then constructed by combining any paraphrase templates from a similarity neighbour with any fillers matching the given part-of-speech tag. The list of all candidates is cut down to a shortlist of 512 paraphrase candidates. These are subsequently ranked by assigning features to each of the candidate paraphrases and scoring them using weights learned in a maximum ranker by optimizing a loss derived from the probability of all candidates that have been mentioned at least two times in the training set in proportion to the probability of all candidates that are not part of the training annotation for that compound at all. (Paraphrases that were named only once are not used for the parameter estimation).

After scoring, determiners are removed from the paraphrase string and duplicates are removed from the list. The generated list is cut off to yield at most 60 items.

2.1 Data Sources

As sources of evidence in the fit (or lack thereof) of a given verb (as a suspected template filler) with the two target words of a compounds, we use data derived from the fifth revision of the English Gigaword[1], tokenized, tagged and parsed with the RASP parsing toolchain (Briscoe et al., 2006), and from Google's web n-gram dataset[2].

[1]Robert Parker, David Graff, Junbo Kong, Ke Chen and Kazuaki Maeda (2011): *English Gigaword Fifth Edition.* LDC2011T07, Linguistic Data Consortium, Philadelphia.

[2]Thorsten Brants, Alex Franz (2006): *Web 1T 5-gram Version 1.* LDC2006T13, Linguistic Data Consortium, Philadel-

To reproduce very general estimates of linguistic plausibility, we built a four-gram language model based on the combined text of the English Gigaword and the British National Corpus (Burnard, 1995), using the KenLM toolkit (Heafield, 2011). On the one hand, free paraphrases are quite unrestricted, which means that the language model helps also in the case of more exotic paraphrases such as (1d) in the first section. On the other hand, many of the more specialized aspects of plausibility such as preposition attachment or selectional preferences for subjects and direct objects can be cast as modeling (smoothed) probabilities for a certain class of short surface strings, for which an n-gram model is a useful first approximation.

Using the grammatical relations extracted by the RASP toolkit, we created a database of plausible verb-subject and verb-object combinations, defined as having a positive pointwise mutual information score.

In a similar fashion, we used a list of verbs and the `morphg` morphological realizer (Minnen et al., 2001) to extract all occurrences of the patterns "N PREP N", "N PREP (DET) N" for noun-preposition-noun combinations, and "N *that* VBZ" as well as "N VBN *by*" for finding typical cases of an active or passive verb that modifies a given noun.

2.2 Ranking features

The following properties used to score each paraphrase candidate (using weights learned by the MaxEnt ranker):

- language model score `lm`
  The score assigned by the 4-gram model learned on the English Gigaword and the BNC.

- pattern type `tp=`*`type`*
  The pattern type (usually the first two 'interesting' tokens from the paraphrase template, i.e., filtering out determiners and auxiliaries). A list of the most frequent pattern types can be found in Table 1.

- pattern weight `pat`
  The pattern weight as the sum of the (neighbour similarity times number of occurrences) contribution from each pattern template.

- linking preposition `prep_`*`prep`*`=`*`type`*
  This feature correlates occurring prepositions (*prep*) to types of patterns, with the goal of learning high feature weights for preposition/type combinations that fit well together. The obvious example for this would be, e.g., that the *of* preposition pattern fits well with *Y_of_X* paraphrases.

- absent preposition `noprep=`*`type`*
  This feature is set when no *X prep Y* or similar pattern could be found.

- subject preference (VBG, VBZ)
  `subj_subj0`, `subj_n_that_vbz`

  object preference (VBN)
  `obj_dobj0`, `obj_n_vbn_by`
  In cases of verbal paraphrases where the compound head is the subject, we can directly check for corpus evidence for the corresponding subject-verb pattern. A similar check is done for verb-object (or verb-patient) patterns in the paraphrases that involve the head in a passive construction.

- frequent/infrequent subject verb (VBG, VBZ)
  `subj_`*`verb`*, `subj_infrequent`
  Some verbs (*belong*, *come*, *concern*, *consist*, *contain*, *deal*, *give*, *have*, *involve*, *make*, *provide*, *regard*, *run*, *sell*, *show*, *use*, *work*) occur frequent enough that we want to introduce a (data-induced) bias towards or away from them. Other verbs, which are more rare, are treated as a single class in this regard (which means that their goodness of fit is mostly represented through the language model and the selectional preference models).

- frequent/infrequent object verb (VBN)
  a similar distinction is made for a list of verbs that often occur in passive form (*appointed*, *associated*, *based*, *carried*, *caused*, *conducted*, *designed*, *found*, *given*, *held*, *kept*, *meant*, *needed*, *performed*, *placed*, *prepared*, *produced*, *provided*, *related*, *taken*)

- co-occurrence of filler with X (other patterns)
  `other_`*`POS`*`_cooc`, `other_`*`POS`*`_none`
  For pattern types where we cannot use one of

phia.

System	isomorphic	non-isom.
SFS	**0.2313**	0.1795
IIITH	0.2309	0.2584
MELODI I	0.1300	**0.5485**
MELODI II	0.1358	0.5360
of+*for* baseline	0.0472	0.8294

Table 2: Official evaluation results + simple baseline

the selectional preference models, we use a model akin to Pado&Lapata's (2007) syntax-based model that provides association scores based on syntactic dependency arc distance.

3 Evaluation Results

The official evaluation results for the task are summarized in Table 2. Two evaluation scores were used:

- **Isomorphic scoring** maps system paraphrases to (unmapped) paraphrases from the reference dataset, and requires systems to produce the full set of paraphrases gathered from Mechanical Turk workers in order to get a perfect score.
- **Nonisomorphic scoring** scores each system paraphrase with respect to the best match from the reference dataset, and averages these scores over all system paraphrases. A system that performs well in nonisomorphic scoring does not need to produce all paraphrases, but will get punished for producing non-reliable paraphrases.

As apparent from the table, systems either score well on the isomorphic score (producing a large number of paraphrases in order to get good coverage of the range of expressions in the reference) or on the non-isomorphic score (producing a smaller number of paraphrases that are highly ranked in the reference). The difference is also apparent in the case of a hypothetical system that produces "Y for X" and and "Y of X" as the paraphrase for any compound (e.g. *bar for wine* and *bar of wine* for *wine bar*). Because these paraphrases occur quite often as most frequent responses, this would yield a high *non-isomorphic* score, but an *isomorphic* score that is very low.

During system development, the relative quality of system paraphrases for each compound was estimated using Maximum Average Precision (MAP) and the total achievable recall (R_{max}) of the generated paraphrase list. Table 3 shows the MAP score (for paraphrases that were listed at least two times) and achievable recall (for all paraphrases). These measures, unlike the official scores, do not attempt to deal with paraphrase variants (e.g. different prepositions for a verbal paraphrase), but are robust and simple enough to give an impression of the quality of the system response.

Compound	closest neighbour	MAP	R_{max}
share holding	withdrawal line	1.000	0.800
union power	community life	1.000	0.750
truth value	accounting treatment	1.000	0.750
amateur championship	computer study	1.000	0.750
government authority	unit manager	1.000	0.680
wine bar	computer industry	0.000	0.040
mammoth task	consumer benefit	0.000	0.040
obstacle course	work area	0.000	0.040
operating system	telephone system	0.000	0.000
deadweight burden	divorce rate	0.000	0.000

Table 3: Best and worst compounds in cross-validation on the training data

As can be seen by looking at the *achievable recall* figures, it is not always the case that all reference paraphrases are in the list that is ranked by the MaxEnt model. In the lower half of table 3, we see that for these cases, the most-similar item selected by the WordNet-based similarity measure is not very close semantically; whether this is the only influencing factor remains to be seen since some of the best-ranked items in the upper half are also abstract concepts with only-somewhat-close neighbours. Future work would therefore have to cover both improvements to the similarity measure itself and to the ranking mechanism used for the reranking of generated paraphrases.

Acknowledgments

The author's work was funded as part of SFB 833 ("*Constitution of Meaning*") by the Deutsche Forschungsgemeinschaft (DFG).

References

Bird, S., Loper, E., and Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Briscoe, E., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*.

Burnard, L., editor (1995). *Users Reference Guide British National Corpus Version 1.0.* Oxford University Computing Service.

Butnariu, C., Kim, S. N., Nakov, P., Seaghdha, D. O., Spakowicz, S., and Veale, T. (2009). SemEval-2010 task 9: The interpretation of noun compounds using paraphrasing verbs and preposition. In *Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

Finin, T. W. (1980). The semantic interpretation of compound nominals. Report T-96, University of Illinois, Coordinated Science Laboratory.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*.

Hendrickx, I., Kozareva, Z., Nakov, P., Séaghdha, D. O., Szpakowicz, S., and Veale, T. (2013). SemEval-2013 task 4: Free paraphrases of noun compounds. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13.

Lauer, M. (1995). Corpus statistics meet the noun compound: some empirical results. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL 1995)*.

Li, G., Lopez-Fernandez, A., and Veale, T. (2010). Ucd-goggle: A hybrid system for noun compound paraphrasing. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*.

Minnen, G., Caroll, J., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Nakov, P. (2008). Noun compound interpretation using paraphrasing verbs: Feasibility study. In Dochev, D., Pistore, M., and Traverso, P., editors, *Artificial Intelligence: Methodology, Systems, and Applications*, volume 5253 of *Lecture Notes in Computer Science*, pages 103–117. Springer Berlin Heidelberg.

Nulty, P. and Costello, F. (2010). Ucd-pn: Selecting general paraphrases using conditional probability. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.

Padó, S. and Lapata, M. (2007). Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. NAACL 2003*, pages 252–259.

Wijaya, D. T. and Gianfortoni, P. (2011). "nut case: what does it mean?": understanding semantic relationship between nouns in noun compounds through paraphrasing and ranking the paraphrases. In *Proceedings of the 1st international workshop on Search and mining entity-relationship data*, SMER '11, pages 9–14.

Wubben, S. (2010). Uvt: Memory-based pairwise ranking of paraphrasing verbs. In *Proceedings of the 5th International Workshop on Semantic Evaluation*.