RUG-1-Pegasussers at SemEval-2022 Task 3: Data Generation Methods to Improve Recognizing Appropriate Taxonomic Word Relations

Frank van den Berg*, **Gijs Danoe***, **Esther Ploeger***, **Wessel Poelman***
Lukas Edman, Tommaso Caselli
Department of Information Science
University of Groningen
{f.l.van.den.berg, g.danoe, e.ploeger.1, w.g.poelman}@student.rug.nl
{j.l.edman, t.caselli}@rug.nl

Abstract

This paper describes our system created for the SemEval 2022 Task 3: Presupposed Taxonomies - Evaluating Neural-network Semantics. This task is focused on correctly recognizing taxonomic word relations in English, French and Italian. We develop various data generation techniques that expand the originally provided train set and show that all methods increase the performance of models trained on these expanded datasets. Our final system outperforms the baseline from the task organizers by achieving an average macro F1 score of 79.6 on all languages, compared to the baseline's 67.4.

1 Introduction

In this paper, we describe our system and approach for the SemEval 2022 PreTENS (Presupposed Taxonomies: Evaluating Neural Network Semantics) shared task (Zamparelli et al., 2022).[1] The aim of this task is to gain a better understanding of the ability of language models to recognize taxonomic relations between two words.

We focus on subtask 1, which is a binary classification task in which a system should predict whether a sentence is valid or not, depending on the taxonomic word relation in a given sentence. We formulate the following research question:

> What are effective data generation approaches in order to improve a language model's ability to recognize appropriate taxonomic word relations?

In our attempt to answer this question, we experiment with multiple approaches: adding new templates, adding new nouns from similar word lists, adding additional hyponyms, inverting templates and using a paraphrasing model to create sentence variations. We use the expanded training data to fine-tune a base English BERT (Devlin et al., 2019) model for the final classification task.

In our approach, we incorporate all three languages for this task: English, Italian and French. Instead of generating additional data for each language and training separate models, we opt to train an English model and translate the Italian and French sentences to English, before predicting the validity labels. We choose this approach in part because several of our data generation methods are not available for French or Italian. We make use of Google Translate, as this is a widely used state-of-the-art general-domain translation system. Our model, trained on the expanded dataset, scores an average F1 score across all languages of 79.6, which is an improvement over the 67.4 baseline score. We find that the best data expansion technique is to combine multiple approaches, where the output of one method is the input for the next. Our ablation experiments show that our paraphrasing method improves scores the most. All code, data and other related files can be found in our GitHub repository.[2]

2 Task description

For the binary classification subtask, the challenge was to predict the acceptability label assigned to each sentence of the test set. The participants were provided with a training set consisting of 5,838 sample sentences and their validity labels, while the test set contained 14,556 sentences. The splits were provided in English, Italian and French, the latter two being slightly adapted translations of the English dataset. These sentences exemplify constructions enforcing presuppositions on the taxonomic status of their arguments A and B, as can be seen in the following examples:

*Contributed equally

[1] https://sites.google.com/view/semeval2022-pretens/home-page

[2] https://github.com/WPoelman/shared-task

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 247 - 254

I like trees, and in particular birches.	1
I like oaks, and in particular trees.	0

A sentence will only get a validity label of 1 when the taxonomic relations are compatible with the sentence construction.

For this task, participants were free to use external resources, with the exception of lexical resources where semantic relationships (including taxonomic ones) are manually marked, such as WordNet (Miller, 1995) or BabelNet (Navigli and Ponzetto, 2012). However, using these lexical resources was allowed for the generation of data, which is part of our approach.

The task of detecting taxonomic word relations has been tried via various approaches. From purely rule-based (Hearst, 1992), to using semantic tree-like resources (Navigli et al., 2011), to adapting pre-trained language models (Atzori and Balloccu, 2020; Chen et al., 2021) or creating hybrid systems (Shwartz et al., 2016; Ravichander et al., 2020). Since this SemEval task is focused on *neural* (language) models, we opt to use BERT and focus mainly on different data generation techniques.

3 System overview

Our main research focus is to explore the effects of different data generation approaches to expand the English training data. This data is then used to train an English BERT model. The unseen Italian and French sentences (to predict) from the test set are first translated into English before we feed them to the model to get their final prediction. We describe these different stages in more detail below.

3.1 Development data split

In order to evaluate our experiments during the development of our system, we created a test set from the original training data. The aim of this set is to replicate the expected official test data characteristics as well as possible. The original training data, as published by the task organizers, consists of seven different templates, distinguished by the words used to describe the relation between two nouns (thus disregarding pronouns). We categorize these in three types of relations:

1. No hypernym relations possible: *I do not like pigs, I prefer animals*
2. Word A is a hypernym of word B: *I like animals, except pigs*
3. Word B cannot be a hypernym of word A: *I like animals, but not pigs*

The first two categories are seen in two templates, while the last category is only relevant to one template. In order to evaluate our techniques, we created our own test set which consists of all sentences with three templates: one for each defined relation. These templates are not present in the training set. Additionally, we filtered nouns that were used with the verbs 'use' and 'met', i.e. verbs that relate to the noun categories 'materials' and 'people'. These nouns were only present in our test set, to ensure that we also evaluate the system on unseen nouns. Additionally, adding the sentences with these nouns means that the test set contains all seven templates, so that there are four overlapping templates with the training set, as this also seems similar to the official test set, where we expected some overlap with training sentences.

3.2 Data generation methods

3.2.1 New templates

First, new templates were added. As described in the previous section, our initial training split consists of only four templates. However, since sentences occur in many different variations, we manually wrote templates in the three previously distinguished relation categories.[3] In total, we added 58 new templates: 22 for the first relation, 23 for the second and 13 for the third.

3.2.2 New words

Our next step was to add new words to the templates. In the task description it was mentioned that the nouns were divided into 30 semantic categories, such as dogs, mammals, motorcycles etc. Not all categories were given, so we tried to infer this from the training data. We extracted all unique nouns from the training data and divided these into lists of different categories. With these categories, we tried to approximate and expand the semantic categories given by the organizers. Certain changes were made, for example, splitting the 'entertainment' category, consisting of books, movies, games and music into separate categories, or combining mammals and dogs into an 'animals' category. Words that occurred together but did not fit into a category, which includes emotions and buildings among others, got assigned to a 'miscellaneous' category. Finally, 14 categories were identified. These categories were also tied to certain verbs in the provided training data. The verb

[3]The full criteria of writing these can be found in our data split description.

'like' was used in all categories, but 'use' was only paired with materials, for example.

The word lists were then enriched. This was mainly done manually, with the help of searching for all hyponyms for a word in WordNet. More verbs were added as well. All categories and their verbs can be found in Table 1. Moreover, we added additional pronouns, with their corresponding possessive pronouns. In addition to *I/my* and *he/his*, we added *she/her* and *they/their*. These lists were used to fill in the existing and new templates. The sentences were checked on their validity using WordNet in order to generate their labels. The generated data is balanced, meaning that each category gets an even amount of relation types, which, in turn, get an even amount of 1's and 0's.

Category	Verbs
Animals	like, **love**
People and professions	like, **love**, met
Materials	like, **love**, use
Games and sports	like, **love**, **enjoy**, **play**
Clothing and jewelry	like, **love**, **wear**
Drinks	like, **love**, **enjoy**, **drink**
Food	like, **love**, **enjoy**, **eat**
Transport	like, **love**, **enjoy**
Movies	like, **love**, **enjoy**, **watch**
Music	like, **love**, **enjoy**, **listen to**
Books	like, **love**, **enjoy**, **read**
Plants	like, **love**
Furniture and household items	like, **love**
Miscellaneous	like, **love**, **enjoy**, **feel**, trust

Table 1: Categories and their corresponding verbs. The bold verbs have been added to the data.

3.2.3 Hyponyms of hyponyms

We noticed that in the valid sentences containing appropriate taxonomic relations, the used arguments always have the same role throughout the data set. As an example, in the valid sentence *I like seafood, except salmon* the argument 'seafood' is a hypernym and the argument 'salmon' is a hyponym. In all other sentences of the training data, the word 'salmon' is also exclusively used as a hyponym, even though it can be a hypernym in a sentence such as *I like salmon, except redfish*.

In line with this example, we created additional sentences where words previously used exclusively as a hyponym, were now used as a hypernym. To do so, we extracted all the hyponyms occurring in the training data and searched WordNet for their direct hyponyms. We then used the first five returned results to generate both valid and invalid new sentences. In creating these additional sentences, we aim to challenge the language model to recognize appropriate taxonomic relations even when the role of an argument alternated between being a subcategory and a supercategory.

3.2.4 Inverting

As mentioned before, we categorized the training data into three possible template relations. For a sentence with the relation 'X is a hypernym of Y', e.g. *I like animals, except pigs*, we know that the sentence is valid because 'animal' is a hypernym of 'pig'. Following this logic, we also know that swapping the arguments to create *I like pigs, except animals*, invalidates the sentence. This process of swapping the arguments in a sentence and (possibly) changing the validity is what we call 'inverting'. Note that not every sentence's validity will change when inverted. When swapping the arguments in the valid sentence *I like jazz more than jewelry* to *I like jewelry more than jazz*, the sentence remains valid. Therefore, we carefully looked at the conditions that make a sentence valid or invalid.[4] We then swapped the two arguments of each sentence and changed the validity label when applicable to create additional data. The language model should learn that the validity also depends on the order of the arguments.

3.2.5 Paraphrasing

In addition to data generation, we also wanted to look at synthetic data. This led us to experiment with a neural paraphrasing model. Specifically we used a fine-tuned 'Pegasus' model from Google (Zhang et al., 2020), originally trained on the task of summarizing and fine-tuned on paraphrasing. We used this model as the final step in our data generation pipeline in order to generate paraphrases of all method combinations. The following is an example the paraphrasing outputs:

> I do not like dogs, I prefer blackbirds.
>
> - I prefer blackbirds, I don't like dogs.
> - I don't like dogs and I like blackbirds.
> - I don't like dogs and I prefer blackbirds.

As the example shows, the differences are not drastic, but do introduce some variation. In order to prevent the paraphrasing model from outputting unrelated sentences, we applied some filtering steps to restrain the output of the model. For instance,

[4] Valid and invalid relations for each template are described in our data split description.

we added length constraints, which ensures that the newly generated sentences were not too short, but neither too long. We allowed at most ten paraphrases per sentence.

3.3 Lemmatization

Since the generated sentences were not all grammatically correct (e.g. 'He like' instead of 'He likes') and because of the fact that the nouns in the original data were plural and the nouns extracted from WordNet were singular, we experimented with lemmatizing the nouns and verbs in the generated sentences. This might remedy the incorrect sentences by equally pre-processing all sentences.

3.4 Translation

By combining the output from our different generation methods, we created an English training set. We used this to fine-tune a pre-trained English language model, namely BERT base provided by Hugging Face.[5] Then, in order to predict the validity labels for Italian and French sentences, we translated these into English in order to process them with the English model. We opted for this approach since several of the data generation methods were not available for French or Italian. For instance, there were no paraphrasing models or easily accessible WordNet-like resources available.

For the translation system, we use Google Translate, as this general-domain transformer-based system is the state-of-the-art. Manual inspection revealed that translation quality seemed sufficient for our purposes.

3.5 System

We experiment with various combinations of the previously mentioned data generation techniques. The detailed results of these experiments are described in the Results section. Our final training dataset was created as follows:

1. Take the existing and new templates (not the ones included our test set)
2. Fill these in with new words from our word lists
3. From the resulting sentences, create additional sentences using the noun hyponyms
4. Add paraphrases of all generated sentences
5. Finally, filter out duplicate sentences

This resulted in dataset of 211,354 sentences, which was used to train our final model. Figure 1 shows an overview of our entire system to get the final predictions.

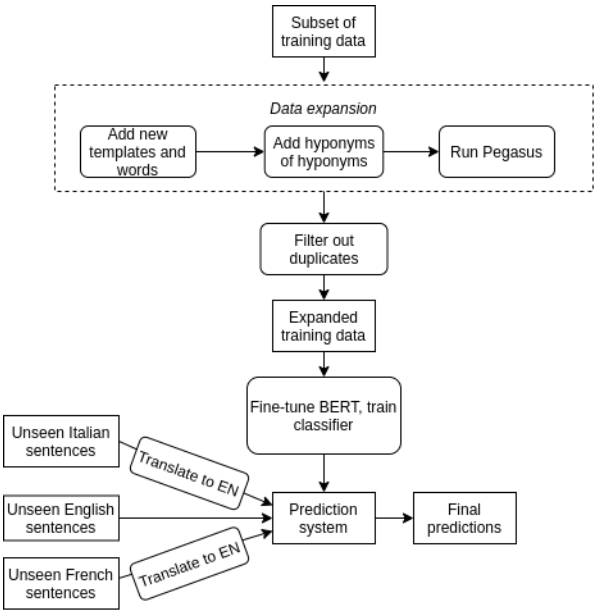

Figure 1: Overview of system and prediction pipeline.

4 Experimental setup

To run our experiments, we created various combinations of our data generation techniques. The full list of experiments can be seen in Table 3. The paraphrasing model we used comes from the Hugging Face model hub.[6]

We trained each system using the Hugging Face transformers library.[7] The exact hyperparameters and other settings can be found in Appendix A. The final model we used for generating our submission can be found in our GitHub repository.

As mentioned, all models were tested using our custom test set with special challenging characteristics.

5 Results

In this section, we discuss the results on both our custom test set and the official test set. We also provide an error analysis in order to gain insight into

[5] https://huggingface.co/bert-base-uncased

[6] https://huggingface.co/tuner007/pegasus_paraphrase

[7] https://huggingface.co/docs/transformers

the contributions of the different data generation methods.

5.1 Custom test set

The full results of our experiments on our custom test set can be found in Appendix B. All data generation methods improve performance over the originally provided training set, while there are some notable differences between the methods. For instance, when looking at the individual methods, we observe that adding the hyponyms has a rather small effect, especially considering that the size of the dataset is almost doubled. The size of the dataset also does not seem be a direct indicator of increased performance. Inverting applicable templates, for example, increases the training set by about 120 sentences, but increases the F1 score from 53.1 to 61.7.

Another surprising effect can be observed in the lemmatized and normal versions of the same datasets. With the individual methods we can see a clear increase in performance, but this effect is not visible with the fully combined methods. The same applies to the inverting method, which resulted in worse performance in some cases in the fully combined methods, which was not the case when it was used on its own.

5.2 Official test set

The official evaluation for the first subtask was twofold:

1. Ranking per language.
2. Global ranking - the average score across all three languages.

While the systems were evaluated using the metrics of precision, recall, micro F1 and macro F1 scores, the official rankings were based only on macro F1 scores.

With our submission, we achieve an average macro F1 score of 79.6, which ranks our system 14th out of 21 participants. We achieve macro F1 scores of 80.3, 78.6 and 79.7 for English, Italian and French respectively. Our system improves over the baseline system, which obtains a macro F1 score of 67.4.

5.3 Error analysis

Once the official test set with labels was available, we conducted ablation experiments to see what effects the different data generation methods had on our submission. Of all methods, the sentences generated with the Pegasus model seem to contribute the most to our final model. The full results of these experiments can be seen in Table 2. To our surprise, the split of our best dataset *without* the new words scores better on the official test set. An explanation could be that using the extensive word lists, which contain rare words, can lead to the generation of vague or uncommon sentences. However, this effect was not apparent in our custom test set, both as an individual method as well as combined with others. Apart from this, the scores are all quite close and again show that the increase in dataset size does not directly lead to better performance.

	Sentences	Acc	Pre	Rec	F1
Templates, new words, hyponyms, pegasus (used for final submission)	211,354	80.7	86.0	70.4	77.4
- pegasus	33,571	77.2	80.0	68.9	74.0
- new words	108,819	81.9	84.1	75.9	79.8
- hyponyms	116,234	79.0	79.9	74.1	76.9

Table 2: Ablation test with final expanded training dataset on official test set.

For each ablation test, we analyze incorrectly predicted examples from a particular model, which were predicted correctly by the others. We look at both the templates and words of the official test set. Additionally, the official set provides information about the structure that was used by the task organizers to generate a particular sentence, which we also include this in the analysis.

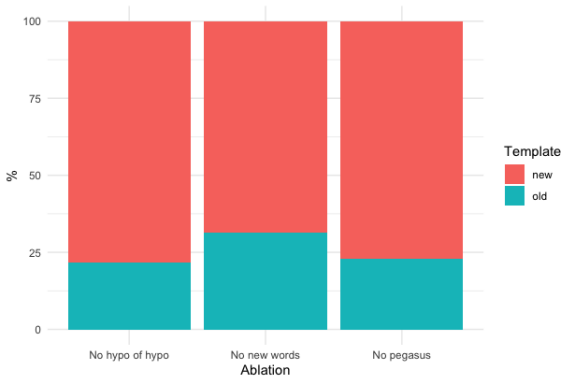

Figure 2: The ratio of incorrectly predicted instances with old and new templates per ablation test.

In Figure 2 we can see the effects of the ablation methods on the templates in the test set. Not adding new words results in a lower error rate for new templates. This is unexpected, as generating sentences with new words was also done using new manual

templates, however the test set still contains completely unseen templates. Another surprise is that not including paraphrasing did not result in relatively more incorrect predictions on new templates, since paraphrasing does introduce some variation in templates to an extent.

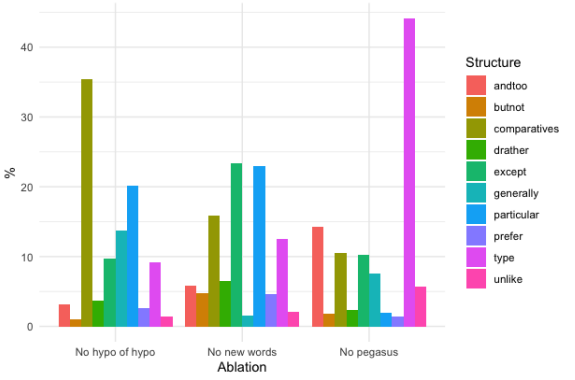

Figure 3: The ratio of structure types in the incorrectly predicted instances per ablation test.

Figure 3 displays the effect of the ablation tests on the different structures. Without the hyponyms of hyponyms method, the error rate of the *comparatives* is high. Excluding the paraphrases results in a higher error rate for the *type*-structure and a lower error rate for the *particular*-structure. This could be because the *type*-sentences are mostly paraphrased correctly which is not the case for *particular*-sentences.

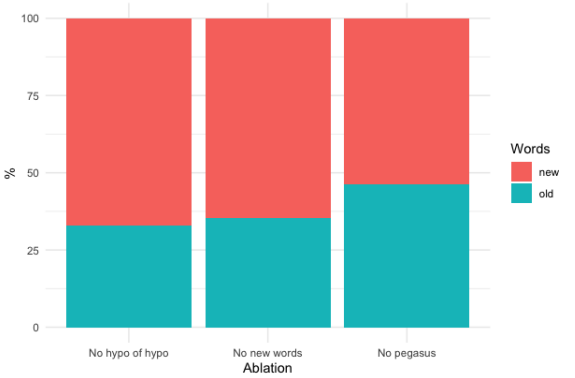

Figure 4: The ratio of incorrectly predicted instances with old and new words per ablation test.

Figure 4 shows the effect of the ablation tests on old and new words. We expected the error rate for ablating new words to be higher. However, new words in the test set were mostly from completely new categories, as opposed to more words from the same categories, which our approach was based on.

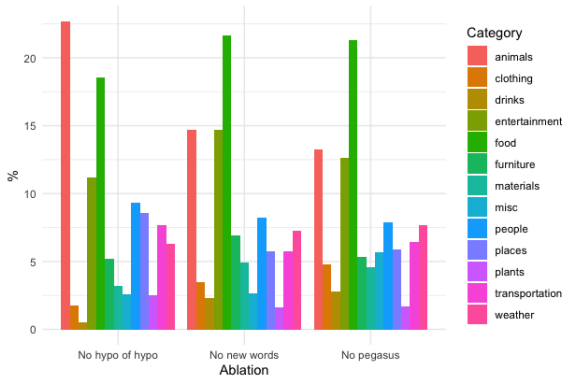

Figure 5: The ratio of word categories in the incorrectly predicted instances per ablation test.

Figure 5 shows the effect on the different categories of words for the ablation tests. These are categorized in the same manner as described in the 'new words' section. There are two new categories in the official test set: *places* and *weather*. Both did not see an increase in error rate by ablating any method. The error rate of *animals* increased by ablating hyponyms of hyponyms. The extensive manner in which the taxonomy of animals is represented in WordNet might be the reason behind this.

6 Conclusion

We have outlined several data generation methods and show that all methods improve the performance of a model trained on the expanded datasets compared to the original training data. Especially new words, new templates and paraphrasing are effective, even by themselves. Surprisingly, adding hyponyms is not that effective by itself, but combined with the previously mentioned methods it scores quite well. The final system we used to participate in the task was trained on a dataset created from those four methods. With this system, we improve the baseline set by the task organizers, placing us 14th out of 21 participants.

As we have shown, some unexpected effects occur when looking at how different data generation techniques perform when combined. In future research, it might be interesting to explore this in a broader context: what can cause such differences? Furthermore, since the provided dataset consisted of short and specific template-based sentences, it could be interesting to experiment with longer sentences that contain more complex constructions.

7 Acknowledgments

We are grateful to Tommaso Caselli and Lukas Edman for their supervision and helpful comments. We also want to thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster.

References

Maurizio Atzori and Simone Balloccu. 2020. Fully-unsupervised embeddings-based hypernym discovery. *Information*, 11(5).

Catherine Chen, Kevin Lin, and Dan Klein. 2021. Constructing taxonomies from pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4687–4700, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence*, 193:217–250.

Roberto Navigli, Paola Velardi, and Stefano Faralli. 2011. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method.

Roberto Zamparelli, Shammur A. Chowdhury, Dominique Brunato, Cristiano Chesi, Felice Dell'Orletta, Arid Hasan, and Giulia Venturi. 2022. Semeval-2022 task3 (pretens): Evaluating neural networks on presuppositional semantic knowledge. In *Proceeding of SEMEVAL 2022*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

A Hyperparameters

We used the Trainer class from Hugging Face, which was mostly left at the default settings. The log of the trainer is included in the folder with our final model, the link to which can be found in the readme of our repository. We put in a time limit of 6 hours for all models. Parameters:

```
optimizer: AdamW
learning rate: 5e-05
batch: 8
scheduler: linear
max epochs: 4
```

B Experiment results on custom test set

	Sentences	F1
Individual methods		
Train	2,737	53.1
Train, hyponyms	4,957	55.5
Train, inverted	2,868	61.7
Train, new words	21,456	65.4
Train, templates (new words)	9,000	71.3
Train, templates (only original train set words)	9,000	73.1
Train, new words lemmatized*	21,456	73.1
Train, pegasus	19,484	77.6
Combined methods		
Train, hyponyms, templates	18,831	70.0
Train, new words, templates	21,456	74.7
Full pipeline combinations		
Templates, new words, inverted, pegasus	138,572	57.9
Templates, new words, hyponyms, pegasus, lemmatized*	211,354	59.1
Templates, new words, hyponyms, inverted	40,820	62.2
Templates, new words, hyponyms, inverted, pegasus	282,834	81.5
Templates, new words, hyponyms, inverted, pegasus, lemmatized*	282,796	83.6
Templates, hyponyms, inverted, pegasus	147,008	85.6
Templates, new words, hyponyms, pegasus	211,354	**88.1**

Table 3: Data experiment results on our own test set. 'Train' refers to the original train set. *Models trained on lemmatized input data were also evaluated on a lemmatized test set. The model trained on the best scoring dataset, in bold, was used for the final submission.