KnCe2013-CORE:Semantic Text Similarity by use of Knowledge Bases

Hermann Ziak
Know-Center GmbH
Graz University of Technology
Inffeldgasse 13/ 6. Stock
8010 Graz, Austria
hziak@know-center.at

Roman Kern
Know-Center GmbH
Graz University of Technology
Inffeldgasse 13/ 6. Stock
8010 Graz, Austria
rkern@know-center.at

Abstract

In this paper we describe KnCe2013-CORE, a system to compute the semantic similarity of two short text snippets. The system computes a number of features which are gathered from different knowledge bases, namely WordNet, Wikipedia and Wiktionary. The similarity scores derived from these features are then fed into several multilayer perceptron neuronal networks. Depending on the size of the text snippets different parameters for the neural networks are used. The final output of the neural networks is compared to human judged data. In the evaluation our system performed sufficiently well for text snippets of equal length, but the performance dropped considerably once the pairs of text snippets differ in size.

1 Introduction

The task of the semantic sentence similarity is to assign a score to a given pair of sentences. This score should reflect the degree by which the two sentences represent the same meaning. The semantic similarity of two sentences could then be used in a number of different application scenarios, for example it could help to improve the performance of information retrieval systems.

In the past, systems based on regression models in combination with well chosen features have demonstrated good performance on this topic[4] [6]. Therefore we took this approach as a starting point to develop our semantic similarity system; additionally, we integrated a number of existing knowledge bases into our system. With it, trained with the data discussed in the task specification of last year[1], we participated in the shared task of SEM 2013.

Additionally, to the similarity based on the features derived from the external knowledge bases, we employ a neural network to compute the final similarity score. The motivation to use a supervised machine learning algorithm has been the observation that the semantic similarity is heavily influenced by the context of the human evaluator. A financial expert for example would judge sentences with financial topics different to non financial experts, if occurring numbers differ from each other.

The remainder of the paper is organised as follows: In Section 2 we described our system, the main features and the neuronal network to combine different feature sets. In Section 3 the calculation method of our feature values is discribed. In Section 4 we report the results of our system based on our experiments and the submitted results of the test data. In Section 5 and 6 we discuss the results and the outcome of our work.

2 System Overview

2.1 Processing

Initially the system puts the sentence pairs of the whole training set through our annotation pipeline. After this process the sentence pairs are compared to each other by our different feature scoring algorithms. The result is a list of scores for each of these pairs where every score represents a feature or part of a feature. The processed sentences are now separated by their length and used to train the neuronal

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task*, pages 138–142, Atlanta, Georgia, June 13-14, 2013.

network models for each length group. The testing data is also grouped based on the sentence length and the score for each pair is determined by a relevant model.

2.2 Token Features

The first set of features are simply the tokens from the two respective sentences. This feature set should perform well, if exactly the same words are used within the pair of sentences to be compared. But as soon as words are replaced by their synonyms or other semantically related words, this feature set will not be able to capture the true similarity. Used without other features it could even lead to false positive matches, for example given sentences with similar content but containing antonyms. The tokenizer used by our system was based on the OpenNLP maximum entropy tokenizer, which detects token boundaries based on probability model.

2.3 Wiktionary Features

While the collaboratively created encyclopedia Wikipedia receives a lot of attention from the general public, as well as the research community, the free dictionary Wiktionary[1] is far lesser known. The Wiktionary dictionary stores the information in a semi-structured way using Wikimedia syntax, where a single page represents a single word or phrase. Therefore we developed a parser to extract relevant information. In our case we were especially interested in semantically related terms, where the semantic relationship is:

Representations: Set of word forms for a specific term. These terms are expected to indicate the highest semantic similarity. This includes all flexions, for example the 's' suffix for plural forms.

Synonyms: List of synonyms for the term.

Hyponyms: List of more specific terms.

Hypernym: Terms which represent more general terms.

Antonym: List of terms, which represent an opposing sense.

Related Terms: Terms, with a semantic relationship, which does not fall in the aforementioned categories. For example related terms for 'bank' are 'bankrupt'. Related terms represent only a weak semantic similarity.

Derived Terms: Terms, with overlapping word forms, such as 'bank holiday', 'bankroll' and 'databank' for the term 'bank'. From all the semantic relationship types, derived terms are the weakest indicator for their similarities.

[1] http://en.wiktionary.org

2.4 WordNet Features

The WordNet[5][2] features were generated identically to the Wiktionary features. We used the WordNet off line database and the provided library to get a broader knowledge base. Therefore we extract the semantically related terms of each token and saved each class of relation. Where each dependency class produced an one value in the final feature score list of the sentence pairs.

2.5 Wikification Feature

We applied a Named Entity Recognition component, which has been trained using Wikipedia categories as input. Given a sentence it will annotate all found concepts that match a Wikipedia article, together with a confidence score. So for every found entry by the annotator there is a list of possible associated topics. The confidence score can then be used to score the topic information, in the final step the evaluation values where calculated as follows:

$$score_{wiki}(s_1, s_2) = \frac{|T_1 \cap T_2|}{norm(T_1, T_2)}$$

where T_1 and T_2 are the set of topics of the two sentences and $norm$ is the mean of the confidence scores of the topics.

2.6 Other Features

Although we mainly focused our approach on the three core features above, others seemed to be useful to improve the performance of the system of which some are described below.

Numbers and Financial Expression Feature: Some sentence pairs showed particular variations between the main features and their actual score. Many of these sentence pairs where quite similar in their semantic topic but contained financial expressions or numbers that differed. Therefore these expressions where extracted and compared against each other with a descending score.

NGrams Feature: The ngram overlapping feature is based on a noun-phrase detection which returns the noun-phrases in different ngrams. This noun-phrase detection is a pos tagger pattern which matches multiple nouns preceding adjectives and determiners. In both sentences the ngrams where extracted and compared to each other returning only the biggest overlapping. In the end, to produce the evaluation values, the word-count of the overlapping ngrams were taken.

3 Distance calculation

For the calculation of the distance of the different features we chose a slightly modified version of the Jacquard similarity coefficient.

$$Jsc(w,l) = \frac{w}{l}$$

Where in this case w stands for the intersection of the selected feature, and l for $\frac{l_a+l_b}{2}$ where l_a and l_b are the length of the sentences with or without stop-words depending on the selected feature. The assumption was that for some features the gap between sentences where one has many stop-words and sentences with none would have a crucial impact but for others it would be detrimental. In regard to this we used, depending on the feature, the words or words excluding stop-words.

3.1 Scoring

One of the main issues at the beginning of our research was how to signal the absence of features to the neuronal network. As our feature scores depend on the length of the sentence, the absence of a particular feature (e.g. financial values) and detected features without intersections (e.g. none of the found financial values in the sentences are intersecting) in the sentence pairs would lead to the same result.

Therefore we applied two different similarity scores based on the feature set. They differ in the result they give, if there is no overlap between the two feature sets.

For a simple term similarity we defined our similarity score as

$$score(w,s,l) = \begin{cases} -1 & : s = 0 \; or \; w = 0 \\ Jsc(w,l) & : w > 0 \end{cases}$$

where w stands for the intersections and S for the word-count of the sentences. The system returns the similarity of -1 for no overlap, which signals no similarity at all. For fully overlapping feature sets, the score is 1.

For other features, where we did not expect them to occur in every sentence, for example numbers or financial terms, the similarity score was defined as follows:

$$score(w,s,l) = \begin{cases} 1 & : s = 0 \; or \; w = 0 \\ Jsc(w,l) & : w > 0 \end{cases}$$

In this case the score would yield 1 decreasing for non overlapping feature sets and will drop to -1 the more features differentiated. This redefines the normal state as equivalent to a total similarity of all found features and only if features differ this value drops.

3.2 Sentence Length Grouping

From tests with the training data we found that our system performed very diversly with both long and short sentences although our features where normalized to the sentence length. To cover this problem we separated the whole collection of training data into different groups based on their length, each of the groups were later used to train their own model. Finally the testing data were also divided into this groups and were applied on the group model.

3.3 Neural Network

We applyied multilayer perceptron neuronal networks on the individual sentence length groups. So for each group of sentence length we computed separately the weights of the neural network. To model the neural networks we used the open-source library Neuroph.[2]. This network was defined with a 48-input layer, which represented the extracted feature scores, 4 hidden layers, and a 1-output layer which represents the similarity score of the sentences. For the runs referenced by table 1 and 2 we used 400000 iterations, which gave us the best results in our tests, with a maximum error of 0.001 and a learning rate of 0.001

[2] http://neuroph.sourceforge.net

4 Evaluation and Results

The following results of our system where produced by our test-run after the challenge deadline. For the first run we split each training set in halfe, self-evident without the use of the datasets published after the challenge, and used the other half to validate our system. See table 1 for result, which contain our system.

	MSRvid	MSRpar	SMTeuroparl
Grouping	0.69	0.55	0.50
Without Grouping	0.66	0.52	0.62

Table 1: Run with and without sentence length grouping on the training set

For the validation the whole 2013 test set was used as it wasnot used for training. In table 2 the results of our system on the test-set are listed. When using the sentence length grouping and without sentence length grouping just using a single neural network for all sentence similarities.

	FNWN	headlines	OnWN	SMT
Grouping	0.08	0.66	0.62	0.21
Without Grouping	0.38	0.62	0.39	0.25

Table 2: Results of our system with and without sentence length grouping on the test set

Finally, we report the results from the original evaluation of the STS-SharedTask in table 3.

	FNWN	headlines	OnWN	SMT
KnCe2013-all	0.11	0.35	0.35	0.16
KnCe2013-diff	0.13	0.40	0.35	0.18
KnCe2013-set	0.04	0.05	-0.15	-0.06

Table 3: The submission to the challenge

5 Discussion

Based on the results we can summarize that our submitted system, worked well for data with very short and simple sentences, such as the MSRvid; however for the longer the sentences the performance declined. The grouping based on the input length worked well for sentences of similar length when compared, as we used the average length of both sentences to group them, but it seamed to fail for sentences with very diverse lengths like in the FNWN data set as shown in table 2. Comparing the results of the official submission to the test runs of our system it underperformed in all datasets. We assume that the poor results in the submission run were caused by badly chosen training settings.

6 Conclusion

In our system for semantic sentence similarity we tried to integrate a number of external knowledge bases to improve its performance. (Viz. WordNet, Wikipedia, Wiktionary) Furthermore, we integrated a neural network component to replicate the similarity score assigned by human judges. We used different sets of neural networks, depending on the size of the sentences. In the evaluation we found that our system worked well for the most datasets. But as soon as the pairs of sentences differed too much in size, or the sentences were very long, the performance decreased. In future work we will consider to tackle this problem with partial matching[3] and to introduces features to extract core statements of short texts.

Acknowledgements

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[1] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor González. Semeval-2012 task 6: A pilot on semantic textual similarity. In *SEM 2012: The First Joint Conference on Lexical and Computational Semantics (SemEval 2012)*, Montreal, Canada, 2012.

[2] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[3] Prodromos Malakasiotis and Ion Androutsopoulos. Learning textual entailment using svms and string similarity measures.

[4] Nikos Malandrakis, Elias Iosif, and Alexandros Potamianos. Deeppurple: estimating sentence semantic similarity using n-gram regression models and web snippets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 565–570, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[5] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[6] Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.