ASAP-II: From the Alignment of Phrases to Text Similarity

Ana O. Alves[1,2]
David Simões[1]
[1]Polytechnic Institute of Coimbra
Portugal
aalves@isec.pt
a21210644@alunos.isec.pt

Hugo Gonçalo Oliveira[2]
Adriana Ferrugento[2]
[2]CISUC, University of Coimbra
Portugal
hroliv@dei.uc.pt
aferr@student.dei.uc.pt

Abstract

ThisThis work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: http://creativecommons.org/licenses/by/4.0/ paper describes the second version of the ASAP system[1] and its participation in the SemEval-2015, task 2a on Semantic Textual Similarity (STS). Our approach is based on computing the WordNet semantic relatedness and similarity of phrases from distinct sentences. We also apply topic modeling to get topic distributions over a set of sentences as well as some linguistic heuristics. In a special addition for this task, we retrieve named entities and compound nouns from DBPedia. All these features are used to feed a regression algorithm that learns the STS function.

1 Introduction

Semantic Textual Similarity (STS), which is the task of computing the similarity between two sentences, has received an increasing amount of attention in recent years (Agirre et al., 2012; Agirre et al., 2013; Marelli et al., 2014a; Agirre et al., 2014; Agirre et al., 2015). Our contribution to this challenge is to learn the STS function for English texts. ASAP-II is an evolution of the ASAP system (Alves et al., 2014), which participated in *SemEval 2014 - Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment*. Although with a different goal from STS, which goes beyond relatedness and entailment, and different datasets, which include pairs of short texts instead of controlled sentences, we believe that, rather than specifying rules, constraints and lexicons manually, it is possible to adapt a system from one to the other task, by automatically acquiring linguistic knowledge through machine learning (ML) methods. For this purpose, we apply some pre-processing techniques to the training set in order to extract different types of features. On the semantic aspect, we compute the similarity/relatedness between phrases using known measures over WordNet (Miller, 1995).

Considering the problem of modeling a text corpus to find short descriptions of documents, we aim at an efficient processing of large collections, while preserving the essential statistical relationships that are useful for similarity judgment. Therefore, we also apply topic modeling, in order to get topic distribution over each sentence set. These features are then used to feed an ensemble ML algorithm for learning the STS function. Our system is entirely developed as a Java independent software package, publicly available[2] for training and testing on given and new datasets containing pairs of texts.

The remainder of this paper comprises 4 sections. In section 2, fundamental concepts are introduced in order to understand the proposed approach delineated in section 3. Section 4 presents some results of our approach, using not only the SemEval-2015's dataset, but also datasets from previous tasks. Finally, section 5 presents some conclusions and complementary work to be done in a near future.

[1] This work was supported by the InfoCrowds project - FCT-PTDC/ECM-TRA/1898/2012

[2] See `https://github.com/examinus-/ASAP`

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 184–189,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

2 Background

2.1 Knowledge Bases

WordNet (Miller, 1995) is a lexical knowledge base structured in synsets – groups of synonymous words that may be seen as possible lexicalizations of a concept – and relations between them, including hypernymy or part-of. DBpedia (Auer et al., 2007) is an effort for extracting structured information from Wikipedia, a well-known collaborative encyclopedia. DBPedia is a central part of the Linked Data initiative and consequently, it is linked to many other resources, including a RDF version of WordNet. In fact, some DBPedia entities are connected to their abstract category in WordNet, through the `wordnet_type` property. For instance, *CNN* is connected to the synset {*channel, transmission channel*} and *Berlusconi* to {*chancellor, premier, prime minister*}.

2.2 Semantic Similarity

There are two main approaches to semantic similarity: (i) semantic relatedness is based on co-occurrence statistics, typically over a large corpus; (ii) classic semantic similarity exploits semantic relations in a lexical knowledge base, such as WordNet. Semantic similarity differs from semantic relatedness because it computes proximity between concepts in a given concept hierarchy (see (Resnik, 1995) and (Jiang and Conrath, 1997)), while the former computes the usage of common concepts together (see (Lesk, 1986), in this case on dictionary definitions/glosses).

2.3 Topic Modeling

Topic modeling relies on the assumption that documents are mixtures of topics, which, in turn, are probability distributions over words. Latent Dirichlet Allocation (LDA) is a generative probabilistic topic model (Blei et al., 2003) where documents are represented as random mixtures over latent topics, characterized by a distribution over words. Assumptions are not made on the word order, only their frequency is relevant. In LDA, main variables are the topic-word distribution Φ and topic distributions θ for each document.

3 Proposed Approach

Our approach to STS is based on a regression function, learned automatically to compute the similarity between sentences, using their components as features. Sentence features are obtained after a preprocessing stage, where sentences are lexically, syntactically and semantically decomposed to obtain different partial similarities. Clustering is applied by LDA in order to obtain the difference of topic distribution between pairs of sentences, which can be considered a composed partial similarity on each topic distribution. Partial similarities are used as features in the supervised learning process. In the following section, complementary stages of our system are explained in detail.

3.1 Natural Language Preprocessing

Sentences are decomposed after applying well-known Natural Language Processing subtasks, namely tokenization, part-of-speech tagging and chunking. For this purpose, we use OpenNLP[3], a tool for processing natural language text out-of-the-box, based on a maximum entropy (ME) approach (Berger et al., 1996). Although OpenNLP offers an English stemmer, this is not sufficient for our approach. Instead, we rely on the lemmatization performed by the WS4J library[4], with some additional heuristics (see section 3.2.3).

3.2 Feature Engineering

Features encode information from raw data that allows machine learning algorithms to estimate an unknown value. We focus on, what we call, *light* features since they are computed automatically, not requiring a specific labeled dataset and we are using already trained models. Each feature is computed as a partial similarity metric, which will later feed the posterior regression analysis. This process is fully automatized, as all features are extracted using OpenNLP and other tools that will be introduced later. For convenience, we set an id for each feature, which has the form $f\#n, n \in \{1..\}$.

[3] See `http://opennlp.sourceforge.net`

[4] A thread-safe, self-contained, Java implementation of some of useful functions over WordNet.See `https://code.google.com/p/ws4j/`

3.2.1 Lexical Features

Some basic similarity metrics are used as features related exclusively with word forms. In this set, we include for each text: the number of stop words, from the Snowball list (Porter, 2001) ($f1$ and $f2$ respectively) and the absolute difference of those counts ($f3 = |f1 - f2|$); the number of those words expressing negation ($f4$ and $f5$ respectively) and the absolute difference of those counts ($f6 = |f4 - f5|$). In addition, we used the absolute difference of overlapping words for each text pair ($f7..10$)[5].

3.2.2 Syntactic Features

The Max Entropy models of OpenNLP were used for tokenization, part-of-speech tagging and text chunking, applied in a pipeline for identifying Noun Phrases (NPs), Verbal Phrases (VPs) and Prepositional Phrases (PPs) of each sentence. Heuristically, these NPs are further identified as subjects if they are in the beginning of sentences. This kind of shallow parser is useful for identifying the syntactic structure of texts. Considering only this property, different features were computed as the absolute value of the difference of the number of NPs ($f11$), VPs ($f12$) and PPs($f13$) for each text pair.

3.2.3 Semantic Features

When possible, suitable WordNet synsets are retrieved for NPs, VPs and PPs of each sentence. These will enable the computation of similarity measures to be used as semantic features. These phrases might be simple words or compounds, language words or named entities, and they might be inflected (e.g. nouns as *electrics* or *economic electric cars* are in the plural form). In order to increase the coverage of named entities, when a word is not in WordNet, we look it up in DBPedia to identify WordNet synset corresponding to its category. Inflected words can also be problematic because WordNet synsets are retrieved by the lemma of their words. Although some WordNet APIs already perform some kind of lemmatization, many situations are not covered. Therefore, to increase the number of words with a suitable synset, the leftmost word of a compound phrase, generally a modifier, is removed until the phrase is empty or a synset is retrieved. If still unsuccessful and the last word ends with an ‘s’, the last character is removed and the word is looked up again.

After retrieving a WordNet sense for each phrase, semantic similarity is computed for each pair, using Resnik (1995) ($f14$), Jiang & Conrath (1997) ($f15$) and the Adapted Lesk metrics (Banerjee and Pedersen, 2003) ($f16$) using WS4j tool, where algorithms in the WordNet::Similarity (Pedersen et al., 2004) Perl package are implemented. For part-of-speech tagged words with multiple senses, the one maximizing partial similarity is selected.

3.3 Distributional Features

The distribution of topics over documents (in our case, short texts) may contribute to model Semantic Similarity since there is no notion of mutual exclusivity that restricts words to be part of one topic only. This allows topic models to capture polysemy. We may thus see topics as natural word sense contexts, as words occur in different topics with distinct “senses”.

Gensim (Řehůřek and Sojka, 2010) is a machine learning framework for topic modeling. It includes several pre-processing techniques, such as stop-word removal and TF-IDF, a standard statistical method that combines the frequency of a term in a particular document with its inverse document frequency in general use (Salton and Buckley, 1988). This score is high for rare terms that occur frequently in a document and are therefore more likely to be significant.

Gensim computes a distribution of 25 topics over texts with or without using TF-IDF ($f17...41$). Each feature is the absolute difference of topic$_i$ (i.e. $topic[i] = |topic[i]_{s1} - topic[i]_{s2}|$). The euclidean distance over the difference of topic distribution between text pairs was used as another feature ($f42$).

3.4 Supervised Learning

WEKA (Hall et al., 2009) is a large collection of machine learning algorithms, written in Java, used for learning our STS function from aforementioned features.

[5] We thank the *SemEval 2014 - Task 1* organizers for providing a Python script for computing baselines available at `http://alt.qcri.org/semeval2014/task1/data/uploads/sick_baseline.zip`, which we used as a different setting for stop word removal (from 0 to 3, 4 different combinations)

One of four approaches is commonly adopted for building classifier ensembles, each focused on a different level of action. Approach A concerns the different ways of combining the results from the classifiers. Approach B uses different models.At feature level (Approach C), different feature subsets can be used for the classifiers, either if they use the same classification model or not. Finally, datasets can be modified so that each classifier in the ensemble is trained on its own dataset (Approach D) (Kuncheva and Whitaker, 2003).

Different methods where applied such as *Voting* (Franke and Mandler, 1992) (Approach A), *Stacking* (Seewald, 2002) (Approach B), and variation of the feature subset used (Approach C). Voting is perhaps a simpler approach, as it selects the class with the largest number of votes. Stacking is used to combine different types of classifiers and demands the use of another learning algorithm to predict which of the models would be the most reliable for each case. This is done with a meta-learner, another learning scheme that combines the output of the base learners. The predictions of base learners are used as input to the meta-learner.

We used WEKA's "Stacking" (Wolpert, 1992) meta-classifier in our *first run*, combining the following base models: three K-Nearest Neighbour (KNN) classifiers ($K = 1$, $K = 3$, $K = 5$) (Aha et al., 1991); a Linear Regression model without an attribute selection method ($-S1$) and default ridge parameter (1.0^{-8}); three M5P classifiers which implement base routines for generating M5 Model trees and rules with a different minimum number of instances ($M = 4$, $M = 10$, $M = 20$) (Quinlan, 1992; Wang and Witten, 1997). The meta-classifier was a M5P classifier with $M = 4$. Other ensembles were added for the *second* and *third runs*:

1. Stacking combining *three* base models: KNN classifier ($K = 1$); Linear Regression model without an attribute selection method ($-S1$) and default ridge parameter (1.0^{-8}); M5P, with $M = 4$, being the meta-classifier[6].

2. Stacking combining *four* base models: KNN classifier ($K = 1$); Linear Regression model without an attribute selection method ($-S1$) and default ridge parameter (1.0^{-8}); ZeroR, a simple rule-based classifier which determines the median similarity score; and Isotonic Regression model. M5P, with $M = 4$, as the meta-classifier.

3. Voting model of the seven classifiers of the *first run*.

Specifically, the *second* and *third run* consisted in the average similarity score produced by the three models presented above, plus the model considered in the *first run*. The only difference between the two runs was that distributional features were not considered in the third run (Approach C).

4 Some Results and Discussion

Although, STS might look similar to *SemEval 2014 - Task 1*, available datasets showed that they are very different from each other. Therefore, we made individual sets of data for training models and for extracting distributional features to evaluate with each target dataset. In *SemEval 2014 - Task 1*, there was only one homogeneous dataset, SICK (Marelli et al., 2014b), with a relatively big amount of entries (5000 for training, 5000 for evaluation) which generally results in better ML outcome. Since answers-forums, answers-students and belief were from new sources, we opted to target these with the same systems, built with most of the available data from previous STS tasks.Table 1 shows that ASAP-II performed better in the SICK dataset, followed by the two datasets that are recurring (images and headlines). Unexpectedly though, the configuration targeting answers-students performed well with only a little difference to the best performance on the headlines, especially if compared to the very low correlation achieved on both answers-forums and belief. Finally, weighted average pearson coefficient was computed considering the size of each evaluation dataset.

5 Conclusions and Future Work

We used complementary features for learning the STS function, which is also part of the challenge of building Compositional Distributional Semantic Models. For this purpose, for each sentence, we extracted lexical, syntactic, semantic and distributional features. On the semantic aspect, we computed the

[6] A Regression Tree using the M5 algorithm (Quinlan, 1992)

	First-run	Second-run	Third-run
answers-forums	0.2304	0.2374	0.2302
answers-students	0.6503	0.7095	0.6719
belief	0.3928	0.3986	0.4342
headlines	0.6614	0.7039	0.7156
images	0.6548	0.7294	0.7250
SICK	0.7200	0.7013	0.7735
Weighted Average	0.57 ± 0.07	0.62 ± 0.08	0.61 ± 0.07

Table 1: Pearson's correlation coefficient for ASAP-II in *SemEval2015-STS*, by dataset, and a simulation of *SemEval2014 - Task 1*, with the same configuration.

semantic similarity and relatedness between phrases using known measures on WordNet, whose "coverage" was increased with the help of DBPedia. We also applied topic modeling to get topic distributions over sets of sentences. All these features were used to feed an ensemble algorithm for learning the STS function. This resulted in a Pearson's r of 0.62 ± 0.08 in our best performance over different datasets.

We are motivated by this participation in STS and intend to participate in further editions, while improving ASAP. To this end, we should: make a deeper analysis of the ensemble, to identify where it can be improved; try to complement the feature set with additional relevant features; explore different topic distributions while varying the number of topics and hopefully maximizing the log likelihood; and assess the impact of each feature.

References

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. 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 385–393, Stroudsburg, PA, USA.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-agirre, and Weiwei Guo. 2013. sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. SemEval-2014.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, CO, June.

David W. Aha, Dennis Kibler, and Marc K. Albert. 1991. Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66.

Ana Alves, Adriana Ferrugento, Mariana Loureno, and Filipe Rodrigues. 2014. Asap: Automatic semantic alignment for phrases. In *SemEval Workshop, COLING 2014, Ireland*, n/a.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 805–810, CA, USA.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Jürgen Franke and Eberhard Mandler. 1992. A comparison of two approaches for combining the votes of cooperating classifiers. In *Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pages 611–614, Aug.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33.

Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 51(2):181–207, May.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86)*, pages 24–26, NY, USA.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. SemEval-2014.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Robertomode Zamparelli. 2014b. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014*.

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

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, PA, USA.

Martin F. Porter. 2001. Snowball: A language for stemming algorithms. Published online.

Ross J. Quinlan. 1992. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the Workshop on New Challenges for NLP Frameworks (LREC 2010)*, pages 45–50, Valletta, Malta.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523.

Alexander K. Seewald. 2002. How to make stacking better and faster while also taking care of an unknown weakness. In C. Sammut and A. Hoffmann, editors, *Nineteenth International Conference on Machine Learning*, pages 554–561.

Yong Wang and Ian H. Witten. 1997. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.