SemAligner: A Method and Tool for Aligning Chunks with Semantic Relation Types and Semantic Similarity Scores

Nabin Maharjan, Rajendra Banjade, Nobal B. Niraula, Vasile Rus
Department of Computer Science, Institute for Intelligent Systems
The University of Memphis, USA
E-mail: {nmharjan,rbanjade,nbnraula,vrus}@memphis.edu

Abstract

This paper introduces a ruled-based method and software tool, called SemAligner, for aligning chunks across texts in a given pair of short English texts. The tool, based on the top performing method at the Interpretable Short Text Similarity shared task at SemEval 2015, where it was used with human annotated (gold) chunks, can now additionally process plain text-pairs using two powerful chunkers we developed, e.g. using Conditional Random Fields. Besides aligning chunks, the tool automatically assigns semantic relations to the aligned chunks (such as EQUI for equivalent and OPPO for opposite) and semantic similarity scores that measure the strength of the semantic relation between the aligned chunks. Experiments show that SemAligner performs competitively for system generated chunks and that these results are also comparable to results obtained on gold chunks. SemAligner has other capabilities such as handling various input formats and chunkers as well as extending lookup resources.

Keywords: Chunk alignment, Chunk semantic relations, Interpretable semantic similarity

1. Introduction

This paper presents a textual chunk alignment method and software tool called SemAligner which can align textual chunks in a given pair of texts (chunked or plain texts). A chunk is a syntactically meaningful unit which typically consists of a single content word surrounded by a group of function words (Abney, 1991). The SemAligner also assigns semantic relation types and semantic similarity scores to the aligned chunks; thus, the proposed SemAligner tool creates a new category of natural language processing tools called semantic aligners. It should be noted that there exist word alignment tools but they do not assign relation types between aligned tokens, a limitation that hampers their usefulness as explained later.

There is an acute need for accurate semantic aligners. For instance, labeling aligned chunks with the underlying semantic relation type and computing semantic similarity scores for them would be extremely useful for explaining or interpreting why two texts are similar or dissimilar. Indeed, existing semantic textual similarity (STS; Agirre et. al. 2015) systems can quantify the similarity between given text-pairs but do not explain in what ways they are similar, related or unrelated.

An explanatory layer would make a big difference in many Natural Language Processing (NLP) applications such as intelligent tutoring system (Graesser et al. 2012; Rus et al. 2013) and student answer evaluation (Rus et. al. 2006; Nielsen 2009). An explanatory layer would transform NLP applications from black boxes into systems that would act intelligently as they would be able to explain their decisions. The organization of Interpretable Short Text Similarity (iSTS) task at SemEval 2015 (Agirre et. al. 2015) highlights the need for such an explanatory layer in STS systems.

We originally developed SemAligner as an entry system in the pilot iSTS shared task. Our tool emerged as the top performing system among the participating systems. However, at that time it only used gold chunks of the given text-pairs; these chunks were provided by the task organizers. We have since extended the tool such that it can work on plain texts by developing two powerful automated chunkers. The first chunker improves an existing, freely available chunker. We fully developed the other chunker based on Conditional Random Fields (CRF), as explained in Section 3. Our experiments, described later, show that the performance of the tool in both system-generated and gold chunk categories is better or competitive to other systems.

The set of semantic relation types is identical to the set used by the iSTS shared task: **EQUI** (chunks are semantically equivalent), **OPPO** (chunks are opposite in meaning), **SPE1/SPE2** (the chunk in the first/second sentence is more specific than the chunk in second/first sentence), **SIMI** (chunks are similar but not EQUI, OPPO or SPE), **REL** (chunks are related but not EQUI, OPPO, SPE or SIMI), **ALIC** (a chunk is not aligned to any other chunk due to 1:1 alignment restriction) and **NOALIC** (the chunk is unrelated and has no alignment). Each alignment is scored between 0 (NOALIC) and 5 (EQUI) (see Section 4 for details).

An example of two texts, their corresponding chunked versions, and the resulting chunk alignment as provided by our SemAligner tool are shown in Figure 1.

S1: Bangladesh building disaster death toll passes 500
S2: Bangladesh building collapse: death toll climbs to 580
S1: [Bangladesh building disaster][death toll][passes][500]
S2: [Bangladesh building collapse][:][death toll][climbs][to 580]
Alignment Output:
1 2 3 ⇔ 1 2 3 // EQUI // 5.0 // Bangladesh building disaster ⇔ Bangladesh building collapse
4 5 ⇔ 5 6 // EQUI // 5.0 // death toll ⇔ death toll
7 ⇔ 8 9 // SIMI // 3.0 // 500 ⇔ to 580
6 ⇔ 0 // NOALI // 0 // passes ⇔ -not aligned-

0 ⇔ 4 // NOALI // 0 // -not aligned- ⇔ :
0 ⇔ 7 // NOALI // 0 // -not aligned- ⇔ climbs

Figure 1: SemAligner output for a given text-pair.

The SemAligner outputs an alignment in the following format: *<S1-chunk id>⇔<S2-chunk id>//<chunk relation type//chunk score//S1 chunk ⇔ S2 chunk*. Unaligned chunks are identified with a 0 position index while aligned chunks are identified as a sequence of token positions in the input sentences.

The SemAligner is customizable and extendible through a number of options that allow the user to configure the behavior of the tool (cf. Section 4). This Java based tool can be used as a standalone application or as a library. It is freely available for research purposes at the SEMILAR - The Semantic Similarity Toolkit's website[1].

2. Related Work

Most semantic similarity methods are geared towards quantifying the similarity between a pair of texts. Works towards interpreting similarity, i.e. providing a justification of why the two texts are similar or dissimilar, are limited but gaining momentum as described next.

Brockett (2007) annotated datasets to indicate alignment of words and phrases. Other related works are word or phrase based alignment models for statistical machine translation (Och et. al., 2004) and word alignment tools. A most recently released tool is the monolingual word-aligner (Shultan et al., 2014) which works at word level but lacks capabilities to assign semantic relation types. In the area of student answer assessment, Nielsen and colleagues (2009) aligned facets/words in student response with concepts in the reference answer for textual entailment. All these previous works focused primarily on the alignment task without attempting to label the semantic relations among the aligned tokens. The first attempt to assign semantic labels to aligned tokens is by Rus and colleagues (2012) who aligned words using greedy and optimal strategies and presented a method to annotate texts with semantic relations such as IDENTICAL and RELATED at word level. More recently, the already mentioned iSTS task at SemEval 2015 (Agirre et. al., 2015) focused on labeling aligned chunks with different semantic relation types and semantic similarity scores thereby providing an explanatory layer to the core semantic similarity task. Our SemAligner tool makes contributions towards the development of such powerful, interpretable STS and other NLP systems.

3. The Chunkers

In order to evaluate our SemAligner tool, we performed alignment experiments on the iSTS data using both gold chunks and system generated chunks. For system generated chunks, we developed a CRF[2] based chunker using both CoNLL-2000[3] shared task training and test data. This data consists of Wall Street Journal corpus: sections 15-18 as training data (211727 tokens) and section 20 as test data (47377 tokens). We generated shallow parsing features such as previous and next words from current word, current word itself, current word POS tag, previous and next word POS tags and their different combinations as described in Sha and Pereira (2003) for building the CRF model.

We evaluated the chunking accuracy of the CRF chunker by comparing it against the gold chunks of iSTS 2015 data: the training and test data sets each consist of 375 pairs of Images annotation data and 378 pairs of Headlines texts. This chunker yielded the highest average accuracies on both the training and test datasets compared to other chunkers which are described next. The accuracies on the training dataset were 86.20% and 68.34% at chunk and sentence level respectively. For the test dataset, the accuracies were 86.81% and 69% at chunk and sentence level, respectively.

We also chunked the input texts using the Open-NLP[4] chunking library (O-NLP). The results are presented in Table 1. The average (of Images and Headlines data) accuracies were 53.04% at chunk level and a modest 9.27% at sentence level for the training dataset. It yielded similar results on test data.

DataSet	Chunker	CL	SL
Training Data			
Headlines	O-NLP	53.74	13.49
	EO-NLP	80.67	59.39
	CRF	**82.60**	**62.56**
Images	O-NLP	52.35	5.06
	EO-NLP	89.13	72.66
	CRF	**89.74**	**74.13**
Test Data			
Headlines	O-NLP	53.88	16.13
	EO-NLP	80.96	60.18
	CRF	**83.32**	**63.23**
Images	O-NLP	52.71	5.33
	EO-NLP	89.30	72.13
	CRF	**90.29**	**74.93**

Table 1 Comparison of chunking accuracies of the various chunkers at chunk level (CL) and at sentence level (SL) using gold chunks from the iSTS 2015 data.

Given the modest performance of the O-NLP chunker, we analyzed its output (i.e. chunks) and added the following rules to merge some of the chunks which resulted in chunks that make more sense and led to significantly better performance.

(a) PP + NP => PP
(b) VP + PRT => VP
(c) NP + CC + NP => NP

For example, EO-NLP chunker merges chunks *[on]* and *[Friday]* to form single PP chunk *[on Friday]* using rule *(a)*. The Extended Open-NLP chunker (EO-NLP) reported 84.9% chunk level and 66.02% sentence level accuracies, respectively, on average on the training dataset. The accuracy on the test data was comparable at 85.13% chunk level and 66.15 sentence level.

Both the EO-NLP and CRF chunkers are available as part of the SemAligner tool.

[1] http://www.semanticsimilarity.org/
[2] https://taku910.github.io/crfpp/
[3] http://www.cnts.ua.ac.be/conll2000/chunking/
[4] http://opennlp.apache.org/cgi-bin/download.cgi

4. The SemAligner Tool

The SemAligner tool can take chunked or plain text-pairs as input. If the input text-pairs are in plain text format, the tool can first detect the chunks using either the EO-NLP or CRF chunkers, described earlier, depending on the user's choice. It should be noted that before performing chunk alignment, the SemAligner preprocesses the text-pairs by performing stopword marking (stopwords are marked to differentiate them from content-words; some rules use this information), lemmatization, POS tagging and Named-Entity recognition using the Stanford CoreNLP Toolkit (Manning et. al. 2014).

Once the chunks are available, the SemAligner relies on a set of rules to align chunks and detect the semantic relation labels. We discuss the rules only briefly here since they are explained in detail in Banjade, Maharjan, Niraula, et al. (2015). There is a subset of alignment rules for each semantic relation type. There are 5 EQUI rules, 1 OPPO rule, 3 SPE rules, 5 SIMI rules, 1 ALIC rule and 1 NOALIC rule. The rules are applied only when certain conditions are met. While aligning chunks, these rules are applied in the following order of precedence: NOALIC, EQUI, OPPO, SPE, SIMI, REL and ALIC. Also, there is a precedence of rules within each relation type. For example, the rule *Both chunks have same tokens (E.g. to compete ⇔ To Compete*) is always applied first before other EQUI rules.

Our SemAligner tool relies on synonym, antonym and hypernym relations in order to align the chunks and therefore use several lookup files to determine these word-to-word semantic relations. All these lookup resources were created using WordNet (Christiane, 1998). There are also rules that use the similarity score between two chunks for determining the alignment. Word to word similarity measures are used to measure chunk to chunk similarity using optimal alignment as described in Stefanescu et al. (2014a). Currently, we use cosine of vectors using the Word2Vec (Mikolov et al., 2013) model as the word-to-word similarity measure as illustrated by the following rule, *if Both chunks have equal number of content words and sim-Mikolov(C1,C2) > 0.6, label as EQUI.* The similarity threshold 0.6 was selected empirically after trying with thresholds varying from 0.4 to 0.9. This rule marks the following two chunks *in Indonesia boat sinking* and *in Indonesia boat capsize* as EQUI.

A chunk can have only one alignment and once aligned, it is not considered for further alignment. Any chunk left unpaired after applying the full set of rules is assigned the NOALIC semantic relation with a score of 0. The aligned chunks with EQUI, OPPO, SPE and ALIC are invariably scored 5, 4, 4 and 0 respectively. The SIMI and REL aligned chunks may have scores between 2 and 4 depending upon the rule being applied. For example, the rule *Each chunk has a token of DATE/TIME type* assigns a score of 3 to the following alignment: *on Friday ⇔ on Wednesday.*

The rules of the SemAligner tool were developed using the training data of iSTS 2015 shared task. Table 2 reports the F1 scores on the training data.

System	A	T	S	T+S
Headline , gold chunks				
SemAligner	0.884	0.639	0.787	0.613
Image , gold chunks				
SemAligner	0.885	0.688	0.800	0.654
Headline , system chunks				
SemAligner	0.821	0.546	0.715	0.523
Image , system chunks				
SemAligner	**0.841**	0.629	0.755	0.599

Table 2: F1 scores on gold and system chunked Headlines and Images training data of iSTS 2015 shared task.

We evaluated the performance of the SemAligner against the gold chunked test data consisting of 378 instances of Headlines and 375 instances of Images datasets used in the iSTS shared task. The system chunks were created using our CRF chunker described in Section 3. The results are presented in Table 3.

System	A	T	S	T+S
Headline , gold chunks				
Baseline	0.844	0.555	0.755	0.555
SemAligner	0.897	**0.666**	0.815	**0.642**
MaxScore	0.898	0.666	0.826	0.642
Image , gold chunks				
Baseline	0.838	0.432	0.721	0.432
SemAligner	0.883	0.603	0.783	0.575
Max Score	0.887	0.614	0.796	0.596
Headline , system chunks				
Baseline	0.670	0.457	0.606	0.4571
SemAligner	**0.826**	**0.564**	**0.736**	**0.543**
Max Score	0.782	0.515	0.702	0.509
Image , system chunks				
Baseline	0.706	0.369	0.609	0.36
SemAligner	**0.852**	0.568	0.749	0.539
Max Score	0.835	0.576	0.751	0.564

Table 3: F1 scores on gold and system chunked Images and Headlines test data. A, T and S refer to Alignment, Type, and Score, respectively. Max Score is the best score for each metric given by any of the participating systems in the shared task.

Our tool performs very well for both gold and system chunks. Our system performs better or competitively in all metric categories versus the best F1 scores (Melamed, 1998) obtained for each metric category among participating systems in the shared task. The SemAligner tool provides the best performance scores (highlighted) across all performance metrics (A, T, S, T+S) in the Headlines dataset with system chunks. For gold chunks in the Headlines dataset, our system performance scores are competitive to the best performance scores across all metrics. Also, the performance scores in the Image dataset (both gold and system chunks) are comparable to the best performance scores of the participating systems in the iSTS task. Interestingly, the performance of our tool using its own chunks (system chunks) is comparable to the results obtained on the gold chunks, showing the general usability of our tool.

The SemAligner has been developed with flexibility in mind. Users can easily customize the application via a configuration file. For example, the user can choose a chunker out of the two. Below are the main configurations to be set for the application.

- *app.input.file* – set full path to text file consisting of tab-delimited text-pairs
- *app.input.format* – set chunked if texts are in chunked form. Otherwise set to plain
- *app.chunking.tool* – set it to either *crf* or *eo_nlp* to select chunking tool. This must be configured if *app.input.format* is set to plain
- *app.out.file* – set valid file path to output file for saving chunk alignment result

The SemAligner tool allows to override default word vector models used in the application. We have only used a subset of vocabulary of pre-trained word2vec model. The vector model can be replaced by user's choice of model. The word model should consists of two files: "**voc.txt**" containing word in each line and "**model.txt**" containing corresponding word vector in each line. The word model format is similar to wiki models developed by Stefanescu et al, 2014b.

- *app.mikolov.models.override* – set it to true for overriding the default word models. Otherwise comment out using # or set it to false
- *app.mikolov.models.path* – set it to folder containing "**voc.txt**" and "**model.txt**" files.

The tool also allows to extend the lookup dictionaries used by the tool.

- *app.extend.lookup.synonym* – extend the synonym dictionary. Each line is a word followed by its tab-delimited synonym words
- *app.extend.lookup.antonym* – extend the antonym dictionary. Each line is a tab delimited word-antonym pair.
- *app.extend.lookup.hypernym* – extend the hypernym dictionary. Each line is word followed by its hypernym and tab delimited.
- *app.extend.lookup.stopword* – extend the stop word list. Each line contains a stop word.

5. Conclusions

This paper introduced a competitive and freely available chunk alignment tool, i.e. SemAligner that can identify semantic relations between the aligned chunks as well as compute semantic similarity scores between the chunks. The SemAligner provides better or comparable performance for both gold and system generated chunked text-pairs. The tool can be very useful for building an explanatory (or interpretable) layer for many NLP applications.

We also plan to release an improved version of the SemAligner tool soon in our website. The improved tool will relax current 1:1 alignment restriction, remove ALIC relation and allow multiple alignments between the chunks.

6. Acknowledgements

This research was partially sponsored by University of Memphis and the Institute for Education Sciences under award R305A100875 to Dr. Vasile Rus.

7. References

Abney, S. (1991). Parsing by Chunks. (pp. 257-278). Springer Netherlands

Agirre, E., & Banea, C. (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), June*.

Banjade, R., Niraula, N. B., Maharjan, N., Rus, V., Stefanescu, D., Lintean, M., & Gautam, D. (2015). NeRoSim: A System for Measuring and Interpreting Semantic Textual Similarity. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), June*.

Brockett, C. (2007). Aligning the RTE 2006 corpus. *Microsoft Research*.

Christiane, F. (1998). WordNet: an electronic lexical database. *Cambrige, MIT Press, Language, Speech, and Communication*.

Corbett, A. T., Koedinger, K. R., & Anderson, J. R. (1997). Intelligent tutoring systems. *Handbook of human computer interaction*, 849-874.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55-60).

Melamed, I. D. (1998). *Manual annotation of translational equivalence* (No. cmp-lg/9805005).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

Nielsen, R. D., Ward, W., & Martin, J. H. (2009). Recognizing entailment in intelligent tutoring systems. *Natural Language Engineering*, *15*(04), 479-501.

Och, F. J., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational linguistics*, *30*(4), 417-449.

Rus, V., & Graesser, A. C. (2006, July). Deeper natural language processing for evaluating student answers in intelligent tutoring systems. In *Proceedings of the National Conference on Artificial INtelligence* (Vol. 21, No. 2, p. 1495). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Rus, V., Lintean, M., Moldovan, C., Baggett, W., Niraula, N., & Morgan, B. (2012). The similar corpus: A resource to foster the qualitative understanding of semantic similarity of texts. In *Semantic Relations II:*

Enhancing Resources and Applications, The 8th Language Resources and Evaluation Conference (LREC 2012), May (pp. 23-25).

Rus, V., D'Mello, S., Hu, X., & Graesser, A. C. (2013). Recent advances in intelligent systems with conversational dialogue. *AI Magazine*, *34*, 42-54.

Rus, V., Lintean, M., Banjade, R., Niraula, N., and Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, August 4-9, 2013, Sofia, Bulgaria.

Sha, F., & Pereira, F. (2003, May). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 134-141). Association for Computational Linguistics.

Ştefănescu, D., Banjade, R., & Rus, V. (2014a). A sentence similarity method based on chunking and information content. In *Computational Linguistics and Intelligent Text Processing* (pp. 442-453). Springer Berlin Heidelberg.

Ştefănescu, D., Banjade, R., & Rus, V. (2014b). Latent semantic analysis models on wikipedia and tasa. In *Language Resources Evaluation Conference (LREC)*.

Sultan, M. A., Bethard, S., & Sumner, T. (2014). Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, *2*, 219-230.