Rev at SemEval-2016 Task 2: Aligning Chunks by Lexical, Part of Speech and Semantic Equivalence

Tan Ping Ping and **Karin Verspoor** and **Tim Miller**
Department of Computing and Information System
University of Melbourne
Australia
{pingt@student., karin.verspoor@, tmiller@}unimelb.edu.au

Abstract

We present the description of our submission to SemEval-2016 Task 2, for the sub-task of aligning pre-annotated chunks between sentence pairs and providing similarity and relatedness labels for the alignment. The objective of the task is to provide interpretable semantic textual similarity assessments by adding an explanatory layer to aligned chunks. We analysed the provided datasets, considering lexical overlap, the part of speech tags and the synonyms of the words in the chunks, and developed a rule-based system reflecting that analysis. Our system performance indicates that when sentence pairs are similar, alignment of chunks can be performed fairly well using lexical information alone without syntactic or semantic analysis. The advantage of our system is that we can easily trace when chunks are aligned.

1 Introduction

We developed a system for SemEval-2016 Task 2: "Interpretable Semantic Textual Similarity" (Agirre et al., 2016), which requires the development of a system that labels aligned chunks of a sentence pair, in terms of their similarity or relatedness. Our approach is based on a detailed analysis of the provided annotation guidelines for the task and the annotated training data.

The annotation guidelines for the task indicate how to align the chunks of two sentence pairs and how to label the similarity or relatedness types and assign scores. Annotations are provided for three different training sets: headlines, images and student responses to questions. Chunks consist of a word or contiguous words. The training sets come in the form of sentence pairs, pre-annotated chunks with their alignment and alignment labels.

The number of sentence pairs for the training and test sets supplied is, respectively: headlines – 726 and 375; images – 750 and 375; and student answers – 330 and 344. The test sets come with the pre-annotated chunks but without alignments or labels. Participants of the sub-task are required to align the pre-annotated chunks and provide the similarity and relatedness assessment for the alignment, which is considered as an explanatory layer that enriches measurement of similarity between the sentence pairs.

The alignment types are semantically equivalent (EQUI), opposite in meaning (OPPO), similar in meaning but specific to the chunk in the first sentence (SPE1), similar in meaning but specific to the chunk in second sentence (SPE2), similar in meaning (SIMI) and related in meaning (REL). Additional types, are factuality (FACT) and polarity (POL). For chunks with no alignment, the label is NOALI. The similarity and relatedness scores are 5 for equivalent (i.e. EQUI//5), [4, 3] for very similar or closely related, [2, 1] for slightly similar or somehow related and 0 for completely unrelated (i.e. NOALI//0).

In order to demonstrate that the relationship between chunks are based on lexical selection, Abney (1991) uses context-free grammar to describe the structure of chunks, providing a definition of a chunk from a linguistic perspective, which he hypothesizes is closer to how humans parse texts. His

Proceedings of SemEval-2016, pages 777–782,
San Diego, California, June 16-17, 2016.

definition provided room for computational implementation. Thus, for this SemEval sub-task, we attempt to understand how lexical overlap, syntactic (more precisely, Part of Speech (PoS) categories), and word synonyms can be used to align chunks, as this information is not provided. Even though string, substring and approximate string matching have been well studied (Yu et al., 2006; Chang and Lawler, 1990; Cole and Hariharan, 2002; Gusfield, 1997), there have been fewer attempts to assess similarity based on not only lexical/character-level overlap but also incorporating other linguistic characteristics. Therefore we attempt to explore the use of this more linguistic information in chunk alignment. We manually examined the annotation guidelines and training set supplied to understand how the chunks are aligned and how the similarity and relatedness labels may be assigned based on this information. Our objective is to develop rules for an automatic system that can effectively align and label the chunks between sentence pairs.

2 Data Analysis

We perform manual data analysis considering lexical matches, part of speech (PoS) categories and synonyms on the headlines training set and derived the rules based on our observation. Through an iterative process, we observed how well the rules derived generalise to the other two training sets (i.e., images and student-answers) and refined the rules. Due to time constraints, the rules were derived mostly from headlines training set and tested on the other two datasets with limited rule refinement iterations from the other two datasets. We summarise our rules below according to the type of lexical match, noting how they correspond to alignment label and score.

[**Punctuations**], such as full stop and comma, which are pre-annotated as individual chunks, do not align.
[**Identical string**] Two identical chunks align directly, and indicate equivalence with relatedness of 5. For example, `injured` $\Longleftrightarrow$ `injured`; `in Iraq` $\Longleftrightarrow$ `in Iraq`. Although the label EQUI//5 is provided, its interpretation as direct lexical matching is confirmed.
[**Sub-string**] Lexical overlap where the chunk from one sentence is a proper sub-string of another generally results in alignment to create similar meaning, with one chunk being more specific than the other (SPE1 or SPE2). For example: SPE1//`gay marriage bill` $\Longleftrightarrow$ `gay marriage`; SPE2//`airstrike` $\Longleftrightarrow$ `new airstrike`.
The following generalisations are also made:

- [Verb and verb agreement] If the PoS tag for the words in the chunks is a verb, direct matching with an addition of 's' on any of the chunks will produce alignment of EQUI//5. For example: `kill` $\Longleftrightarrow$ `kills`; `recognizes` $\Longleftrightarrow$ `recognize`. This captures the situtation that not all EQUI//5 cases are direct matching; in this case the verb is aligned and the variation reflects only verb agreement.

- [Plural and singular] If the PoS for the words in both the chunks is a noun, direct lexical overlap, with an addition of 's' on each side will produce alignment SIMI//4. For example: `Suicide bomber` $\Longleftrightarrow$ `Suicide Bombers`; `despite concession` $\Longleftrightarrow$ `despite concessions`. This occurs when the chunks are the same noun, but plural/singular variants.

- [Syntactic function 'to'] Direct matching of 'to' on either side of the chunks will create alignment of EQUI//5. For example: `to vote` $\Longleftrightarrow$ `vote`; `to NZ same-sex marriage` $\Longleftrightarrow$ `New Zealand same-sex marriage`

[**Number**] For chunks containing numbers, several subcases apply.

- [Same value]. Two chunks containing the same number results in EQUI//5. For example: `at 91` $\Longleftrightarrow$ `aged 91`; `Boeing 787 Dreamliner` $\Longleftrightarrow$ `on 787 Dreamliner`

- Otherwise, chunks with digits are aligned to produce alignment label of SIMI. The Relatedness score depends on the differences of the digits extracted from the chunks. We have identified the following heuristics:

- If the difference is greater than 100, alignment will produce Relatedness of 1 (e.g., `to 35 years` ⟺ `to 1,000 years`)
- If the difference is between 30 and 100, assign Relatedness 2 (e.g., `At least 45` ⟺ `At least 13`)
- If the difference is between 7 and 10, assign Relatedness 3 (e.g., `17` ⟺ `10`)
- if the difference is less than 7, assign Relatedness 4 (e.g., `17` ⟺ `15`)

[**Single word match**] Generally, if a single word in a chunk is string equivalent to a word in another multi-word chunk, they will align. For example: `in bus accident` ⟺ `in road accident`. The following related generalisations are also made:

- [Synonymous] If the two chunks contain a synonymous word rather than an identical word, the chunks are aligned and given labels of EQUI//5. For example: `China stocks` ⟺ `Chinese stocks`; `From Soccer` ⟺ `from football`.

- [PP - modifier to head noun] If the words in the chunks are prepositional phrases (PP) with the same head noun but different modifiers, the chunks are aligned with a label of SIMI//4. For example: `as legitimate representative` ⟺ `as sole representative`.

- [NP - different head noun] If both the words in the chunks are noun phrases (NP) and the matching word is an adjective, the chunks are aligned as REL. For example: `economic traps` ⟺ `economic growth`; `French train` ⟺ `French train passengers`.

Through stemming, individual words in the chunks are converted to root word.

[**Same root word**] Chunks that can be converted to same root word, are aligned as EQUI//5. For example: `detained` ⟺ `detains`; `summoned` ⟺ `summons`.

[**Synonymous root word**] If the root words are synonymous, aligned as EQUI//5 (for example: `to permit` ⟺ `Allowed`)

[**Misc**] A few recurrent cases appeared in training set like OPPO//5 (e.g.: `higher` ⟺ `lower`) and OPPO//4 (e.g.: `close` ⟺ `open`); these chunks are matched as antonyms.

3 Methodology

In order to understand the effect of lexical overlap in aligning chunks between sentences, we developed two separate systems: LexiM and Rev, although we only submitted one run for this sub-task, which is the Rev system. Both systems are implemented in Java and strictly rule-based systems. LexiM, a system purely based on lexical overlap contained 13 rules. The rules are based on string or sub-string matches, which follow the category of our data analysis but excluding cases that considers PoS or word synonym, or the rules that require conversion to root word. We performed error analysis on the other training sets but the overall performance is compared to the baseline approach supplied by the organiser (Table 1).

Rev extends LexiM by adopting the rules of LexiM as outlined in Section 2 (excluding the Misc rule), with addition of string distance and semantic similarity-based strategies. LexiM works to align chunks which have lexical overlap, while string similarity and semantic distance work in two ways. The first way is to align chunks that that cannot be handled with those rules and follow by assining labels to the aligned chunks. The second way is for the chunks that are aligned through LexiM, the string distance and semantic similarity rule will provide similarity and relatedness labels. There are existing tools which perform tasks such as string distance measurement like SecondString (Cohen et al., 2003) and stringmetric (Madden, 2013), PoS taggers like Stanford Parser (Klein and Manning, 2003), and dictionary for synonymous words like WordNet (Fellbaum, 1998). Cohen et al. (2003) shows that Jaro-Winkler is an effective string distance metric for name matching task. Liu et al. (2010) has shown that TESLA, a similarity metric that considers both PoS tags and semantic equivalence (based on WordNet synsets), is effective in the task of auto-

matic evaluation of machine translation in English language. We used Jaro-Winkler proximity (SecondString implementation) for string distance measurement, and TESLA for semantic similarity measurement.

We analyzed the chunks in terms of their string distance and semantic similarity scores, relating the scores to the similarity and relatedness types and scores in the annotated data. We identified the lowest, average and highest values for the string distance and semantic similarity measures for each association type and score, for example: REL//1 corresponds to {string distance: 0.0, 0.19, 0.43} and {semantic similarity: 0.04, 0.18, 0.5}. Subsequently, we developed rules for alignment using string distance and semantic similarity where both ranges must be satisfied in order for the rules to be applied. Chunks that do not fulfill any of the rules or similarity criteria will be assigned as NOALI//0.

4 Results and Discussion

The evaluation method determined by the organiser is F_1-score (Powers, 2011). As shown in our results tables (Table 1 and 2), F_1-score is calculated separately for the aligned chunks (Ali), the alignment type (Type), the similarity and relatedness score (Score) and combination of alignment type and score (Typ+Sco).

The performance of LexiM helps to understand the effect of lexical overlap and sub-string matches in aligning chunks between sentence pairs. We compared the results to the baseline approach (Agerri et al., 2014) provided (Table 1, the Student-Answers baseline is not provided (NS)). The baseline approach is a multilingual Natural Language Processing (NLP) tool that can perform tokenization and segmentation, statistical POS tagging and lemmatization, Named Entity Recognition tagger, probabilistic chunker and constituent parser for several languages, including English. LexiM, although consists of lexical overlap and sub-string matches rules, outperformed the baseline (Table 1). This demonstrates that alignment of chunks can be performed purely using lexical matching; the lexical information was useful for identifying overlaps between the chunks, especially in the case of similar sentences.

Some of the examples where LexiM produces correct alignment with the wrong type and score: `a baby` ⟺ `holding a baby`; `sitting` ⟺ `sit`; `is not connected` ⟺ `are not connected`; `at the negative connection.` ⟺ `the battery.` There are cases where the alignment by LexiM is interpretable. For example in the images dataset, LexiM produced SPE1//4//`in front of yellow flowers` ⟺ `in a field`, while gold standard labels is SIMI//2. Another case, LexiM produced SPE2//4//`sheep` ⟺ `A sheep` while the standard labels are EQUI//5. Even if only two sentences with the chunks are provided, the system is able to produce a reasonable alignment of the chunks.

Dataset	Baseline	LexiM	Rev
Headlines	F_1-Score	F_1-Score	F_1-Score
Ali	0.8354	0.8519	0.8573
Type	0.5392	0.5494	0.5611
Score	0.7409	0.7638	0.7691
Typ+Sco	0.5391	0.5352	0.5464
Images	F_1-Score	F_1-Score	F_1-Score
Ali	0.8364	0.7949	0.8318
Type	0.4444	0.4636	0.4951
Score	0.7186	0.6711	0.7487
Typ+Sco	0.4442	0.4636	0.4854
Student-Answers	F_1-Score	F_1-Score	F_1-Score
Ali	NS	0.7764	0.8513
Type	NS	0.3118	0.3952
Score	NS	0.6210	0.7232
Typ+Sco	NS	0.3060	0.3879

Table 1: Results on Different Training Sets

We observed slight improvement in F_1-scores between LexiM and Rev. The difference between LexiM and Rev is additional lexical information like PoS and synonym. We performed error analysis on the output of LexiM and Rev with comparison to the baseline approach which uses probabilistic approach. We performed error analysis to understand this better: Rev produced correct alignment and labels EQUI//5//`sleeps` ⟺ `asleep` considering the both words are synonymous, while LexiM is able to align the chunks lexically with wrong labels and the baseline approach fails to align the chunks; Rev is able to align `standing` ⟺ `grazing` but with wrong labels, while LexiM and the baseline approach fails to align the chunks. This shows that adding additional information like PoS and syn-

Testing Set	F_1-Score	Rev	Baseline
Headlines	Ali	0.8662	0.8462
(Rank 14 out of 20)	Type	0.5705	0.5462
	Score	0.7844	0.761
	Typ+Sco	0.5624	0.5461
Images	Ali	0.831	0.8556
(Rank 17 out of 20)	Type	0.5014	0.4799
	Score	0.7399	0.7456
	Typ+Sco	0.4929	0.4799
Student-Answers	Ali	0.8458	0.8203
(Rank 18 out of 19)	Type	0.4179	0.5566
	Score	0.7265	0.7464
	Typ+Sco	0.4104	0.5566

Table 2: Official Results

onym can assist in alignment and labelling of the aligned chunks.

The overall performance of Rev using the test sets is presented in Table 2. As stated previously, the rules are derived mostly from the headlines dataset. Therefore it is unsurprising that Rev exhibits the best performance over the headlines test set. Indeed, it performs slightly better on the headlines test set than on the corresponding training set; although the reasons for this are unclear, it suggests that the rules captured in Rev have indeed captured reasonable generalizations.

Unfortunately, Rev did not perform well for the other two datasets because the rules are too specific to the headlines dataset. There are many more other potential rules which could be derived from the other datasets; more targeted effort in rule development for those data sets would improve the performance of Rev. It is also clear from the results that the current rules in Rev are quite brittle; they did not perform well in comparison to other submissions for the task.

Using the chunk alignments produced by Rev, we can easily provide an explanation for the alignment by highlighting the rule that facilitated the alignment. Here are some examples of this, concentrating on cases without direct lexical matching:

[Synonymous root word] `a set of stairs` $\Longleftrightarrow$ `steps`.

[Synonymous root word] `jumps` $\Longleftrightarrow$ `leaps`.

[Number] `22 Dead` $\Longleftrightarrow$ `89 dead`.

[Plural and singular] `The dog` $\Longleftrightarrow$ `Three dogs`.

The weakness of Rev is that the rules are limited to the direct observations of the training set, and only consider a narrow set of linguistic characteristics. Clearly, other linguistic rules could be incorporated. For instance, an antonym dictionary could be consulted; this would directly benefit the OPPO label included in the task.

5 Conclusion

Based on the observation that pairs of chunks that align are often lexically very similar, we have concentrated on alignment using a lexical approach. Although deriving and iteratively refining rules manually is a time consuming process, this process is helpful to build an interpretable semantic textual similarity system.

As future work, we plan to directly include an indication of the rule that produced the alignment as an additional explanation in the output, beyond the similarity and relatedness scores and types alone. There are many other types of linguistic analysis that can be performed and we will consider incorporating these as future enhancements to our system.

References

Steven P Abney. 1991. *Parsing by chunks*. Springer.

Rodrigo Agerri, Josu Bermudez, and German Rigau. 2014. Ixa pipeline: Efficient and ready to use multilingual nlp tools. In *LREC*, volume 2014, pages 3823–3828.

Eneko Agirre, Aitor Gonzalez-Agirre, Iigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016. Semeval-2016 task 2: Interpretable semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June.

William I Chang and Eugene L Lawler. 1990. Approximate string matching in sublinear expected time. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 116–124. IEEE.

William W Cohen, Pradeep D Ravikumar, Stephen E Fienberg, et al. 2003. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, volume 2003, pages 73–78.

Richard Cole and Ramesh Hariharan. 2002. Approximate string matching: A simpler faster algorithm. *SIAM Journal on Computing*, 31(6):1761–1782.

Christiane Fellbaum. 1998. A semantic network of english verbs. *WordNet: An electronic lexical database*, 3:153–178.

Dan Gusfield. 1997. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge university press.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2010. Tesla: Translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 354–359. Association for Computational Linguistics.

Rocky Madden. 2013. stringmetric @ONLINE.

David Martin Powers. 2011. Evaluation from precision, recall and f-measure to roc, informedness, markedness and correlation.

Fang Yu, Zhifeng Chen, Yanlei Diao, TV Lakshman, and Randy H Katz. 2006. Fast and memory-efficient regular expression matching for deep packet inspection. In *Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*, pages 93–102. ACM.