Sentence Alignment using Unfolding Recursive Autoencoders

Jeenu Grover
IIT Kharagpur
India - 721302
groverjeenu@gmail.com

Pabitra Mitra
IIT Kharagpur
India - 721302
pabitra@cse.iitkgp.ernet.in

Abstract

In this paper, we propose a novel two step algorithm for sentence alignment in monolingual corpora using Unfolding Recursive Autoencoders. First, we use *unfolding recursive auto-encoders* (RAE) to learn feature vectors for phrases in syntactical tree of the sentence. To compare two sentences we use a similarity matrix which has dimensions proportional to the size of the two sentences. Since the similarity matrix generated to compare two sentences has varying dimension due to different sentence lengths, a dynamic pooling layer is used to map it to a matrix of fixed dimension. The resulting matrix is used to calculate the similarity scores between the two sentences. The second step of the algorithm captures the contexts in which the sentences occur in the document by using a dynamic programming algorithm for global alignment.

1 Introduction

Neural Network based architectures are increasingly being used for capturing the semantics of the Natural Language (Pennington et al., 2014). We put them to use for alignment of the sentences in monolingual corpora. Sentence alignment can be formally defined as a mapping of sentences from one document to other such that a sentence pair belongs to the mapping iff both the sentences convey the same semantics in their respective texts. The mapping can be many-to-many as a sentence(s) in one document could be split into multiple sentences in the other to convey same information. It is to be noted that this task is different form paraphrase identification because here we are not just considering the similarity between two individual sentences but we are also considering the context in a sense that we are making use of the order in which the sentences occur in documents.

Text alignment in Machine Translation (MT) tasks varies a lot from sentence alignment in monolingual corpora as MT tasks deal with bilingual corpora which exhibits a very strong level of alignment. But two comparable documents in monolingual corpora, such as two articles written about a common entity or two newspaper reports about an event, use widely divergent forms to express same information content. They may contain paraphrases, alternate wording, change of sentence and paragraph order etc. As a result, the surface-based techniques which rely on comparing the sentence lengths, sentence ordering etc. are less likely to be useful for monolingual sentence alignment as opposed to their effectiveness in alignment of bilingual corpora.

Sentence alignment finds its use in applications such as plagiarism detection(Clough et al., 2002), information retrieval and question answering(Marsi and Krahmer, 2005). It can also be used to generate training set data for tasks such as text summarization.

2 Related Work

A lot of work has been done on the problem of sentence alignment which relies on the surface properties of the text in natural language such as word overlap(Hatzivassiloglou et al., 2001; Barzilay and Elhadad, 2003), bag-of-words model(Nelken and Shieber, 2006). It relies mainly in the field of statistical machine learning (Barzilay and Elhadad, 2003). A little has been done to improve upon this task by capturing the semantics of the text.

Barzilay and Elhadad show that a similarity measure combined with contextual information outperforms methods based on sentence similar-

Proceedings of the 10th Workshop on Building and Using Comparable Corpora, pages 16–20,
Vancouver, Canada, August 3, 2017. ©2017 Association for Computational Linguistics

ity functions. Nelken and Shieber improved upon the sentence similarity function by borrowing TF-IDF based scoring from the information retrieval literature and outperformed all other methods.

Their work can be summarized in 4 steps:

1. TF*IDF : Treat each sentence as a document and compute it's TF*IDF vector. For a word $\mathbf{t}$ in sentence[1] $\mathbf{s}$, $TF_s(t)$ denotes the number of times $\mathbf{t}$ occurs in $\mathbf{s}$, $\mathbf{N}$ is the number of sentences in document and $DF(t)$ indicates the occurrences of t in document.
$$w_s(t) = Tf_s(t) \times \log \frac{N}{DF(t)} \tag{1}$$
where $w_s(t)$, denotes the value for dimension corresponding to word t in TF-IDF vector of sentence s.

2. The previous step gave the similarity measure of 2 sentences. It was converted to an appropriate probability measure denoting the $Pr(align(s_i, s_j) = 1)$ by using logistic regression on the training data.

3. Heuristic Alignment : They simply choose sentence pairs between two documents with $pr(align) > th$,where th is the threshold. Additionally heuristics such as mapping the first sentences of two documents (as justified by Quirk et al.(Dolan et al., 2004)) and allowing 2-to-1 mapping of adjacent sentences are followed.

4. Global Alignment with Dynamic Programming: They compute the optimal alignment between sentences 1..i of one text and sentences 1..j of the elementary version by using a dynamic programming approach similar to Needleman and Wunsch (1970).

3 Approach

In this section, we briefly visit the neural network models and other techniques that would be used in our task.

3.1 Neural Embeddings

The idea of using neural embeddings is to get $\mathbf{n}$-dimensional space representations for the words in vocabulary $\mathbf{V}$. We define a mapping
$$\mathbf{L_w} : \mathbf{V} \to \mathbb{R}^{\mathbf{n}} \tag{2}$$
which embeds words into a semantic vector space where the metric approximates semantic similarity. The idea of neural embeddings was first introduced by Bengio et al.(2003) and later worked upon by Turian et al.(2010). Mikolov et al.(2013) points that the words with similar meaning are mapped closer in this new feature space. The directions in the vector space correspond to different semantic concepts.

Turian et al.(2010) gave us an encoding from a given word to a vector in the semantic space. Now, we want to have an embedding from a sentence to a vector in the semantic space, i.e. given,
$$\mathbf{L_w} : \mathbf{V} \to \mathbb{R}^{\mathbf{n}} \tag{3}$$
we want to get,
$$\mathbf{L_s} : \mathbf{V}^* \to \mathbb{R}^{\mathbf{n}} \tag{4}$$

To get such a mapping, we use autoencoders recursively on the parse tree representation of the sentence. Each node in the parse tree represents a vector of dimension $\mathbf{n}$ corresponding to that word or phrase in the sentence.

3.2 Unfolding Recursive Autoencoders with Dynamic Pooling

Socher et al.(2011) first used Unfolding Recursive Autoencoders with dynamic pooling for the purpose of paraphrase identification. We would be using their method in our paper for sentence alignment. We learn the embeddings of all the phrases in the parse tree of the sentences using unfolding RAE. For a given sentence with $\mathbf{N}$ words, we have total $\mathbf{2N-1}$ nodes in the parse tree of the sentence, $\mathbf{N}$ for the words and $\mathbf{N-1}$ for the internal nodes or phrases in the sentence as determined by the parsing of the sentence.

For computing the similarity matrix for two sentences, the rows and columns denote the words in their original sentence order. We then add to each row and column the nonterminal nodes of the parse tree in a depth-first and right-to-left order.

For a sentence with N words, and with word embeddings $x_{1:N}$ and RAE encoding for phrases $y_{1:N-1}$, form
$$s = [x_1, ..., x_N, y_1, ..., y_{N-1}] \tag{5}$$

For two sentences (s_1, s_2), the similarity matrix S contains the Euclidean distance between $(s_1)_i$ and $(s_2)_j$.
$$(S)_{i,j} = \|(s_1)_i - (s_2)_j\|^2 \tag{6}$$

[1] We are using terms ”word” and ”sentence” in their literal sense and not according to the TF-IDF terminology.

For sentence s_1 of size n and sentence s_2 of size m, the matrix has dimension $(2n-1) \times (2m-1)$. Since the resulting similarity matrix has dimension which depends on the lengths of the given sentences, we would use dynamic pooling to convert it into a matrix of fixed dimension.

We would be using dynamic min-pooling to convert the variable sized matrix into a matrix of size $n_p \times n_p$. As Socher et al.(2011) reported, the best suited size for n_p is 15. For dynamic pooling, we divide each dimension of 2D matrix into n_p chunks of $\left\lfloor \frac{len}{n_p} \right\rfloor$ size, where len is the length of dimension. If the length len of any dimension is lesser than n_p, we duplicate the matrix entries along that dimension till len becomes greater than or equal to n_p. If there are l leftover entries where $l = len - n_p * \left\lfloor \frac{len}{n_p} \right\rfloor$, we distribute them to the last l chunks. We do it for both the dimensions.

We are using min-pooling because closer the two phrases are, lesser is the euclidean distance between them. Min-pooling would be able to capture this relationship if there are two phrases in the window which are closer to each other.

3.3 Alignment using similarity scores

The fixed dimension matrix obtained in the previous step was fed to the softmax classifier to get a confidence score about similarity between sentences. We would use a dynamic programming algorithm to find the optimum alignment of sentences between the documents. This approach relies on the document comparability and linearity of sentence ordering in the two documents (albeit weak). We find the maximum optimum alignment between two documents and then backtrack using the alignment matrix M to find the sentences that were aligned. Here, $M(i,j)$ denotes the maximum alignment between sentences $1..i$ of one document to sentences $1..j$ of the other document and $sim(i,j)$ denotes the confidence score as given by softmax classifier for similarity between sentences i and j of the two documents respectively. The $offdiag$ constant is used to skip a match between two sentences if the similarity between them is very low. The value of $offdiag$ constant was chosen to be 0.1 for our experiment.

$$M(i,j) = \max \begin{cases} M(i-1,j-1) + sim(i,j) \\ M(i-1,j) + offdiag \\ M(i,j-1) + offdiag \end{cases} \tag{7}$$

4 Experiment

We would list below the detailed steps of our experiment,

4.1 Unfolding RAE's training

We used a pre-trained model of RAE's as given by Socher et al.(2011) which is trained using a subset of 150,000 sentences from the NYT and AP sections of the Gigaword corpus. They used Stanford parser(De Marneffe et al., 2006) to create the parse trees for all sentences. 100-dimensional vectors computed via the unsupervised method of Collobert and Weston (Collobert and Weston, 2008) and provided by Turian et al.(Turian et al., 2010) were used. The RAE used had two encoding layers. The size of hidden layer used is 200 units.

4.2 Softmax Classifier

For training the softmax classifier to get the similarity scores between two sentences, we used the dataset for similar task i.e. Paraphrase Identification for training as both the tasks are similar when only individual sentences irrespective of their context are considered. Microsoft Research paraphrase corpus (MSRPC) consists of 5801 pairs of sentences which have been extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship.All sentences are labeled by two annotators who agreed in 83% of the cases and third annotator resolved the conflicts. A total of 3,900 sentence pairs are labeled as paraphrases. We used the standard split of 70-30 for training and testing.

4.3 Dataset

For testing our algorithm we took articles literacynet archives[2]. It maintains a collection of stories from CNN and CBF5. The material is intended to be used for promoting the literacy. Each story in the archive has an abridged or shorter version. We

[2] http://literacynet.org

took 5 such pairs of stories and their abridged versions leading two 2033 sentence pairs that could potentially be aligned. We manually annotated the dataset to find the ground truth. The alignment diversity measure (ADM) for two texts, T_1, T_2, is defined to be:

$$ADM(T_1, T_2) = \frac{2 \times matches(T_1, T_2)}{|T_1| + |T_2|} \quad (8)$$

where $matches$ denote the actual number of aligned sentence pairs between two documents. Intuitively, for closely aligned document pairs, as prevalent in bilingual alignment or MT tasks, one would expect an ADM value close to 1. The average ADM in our dataset is 0.61.

4.4 Algorithm

1. Given two texts T_1, T_2, we split each into its sentences. For all sentences s_i in T_1 and for all sentences s'_j in T_2, we generate the embedding vectors for all the words and phrases in the sentences using unfolding RAE.

2. The similarity matrix S is generated for s_i and s'_j by taking Euclidean distance of between all the possible words and phrases of both the sentences as mentioned earlier.

3. Each similarity matrix is converted to fixed size matrix S_{pooled} by using dynamic Min-pooling and is fed to softmax classifier which assigns the confidence score of the two sentences being similar. Now, we have matrix P for all the sentence pairs in T_1 and T_2 such that $P_{i,j}$ represents a measure of similarity between s_i in T_1 and s'_j in T_2.

4. Let $M_{i,j}$ denote the maximum similarity score obtained by aligning the sentences $s_{1:i}$ of T_1 with sentences $s'_{1:j}$ of T_2. We then use a dynamic programming algorithm to maximize this score. We also store the choices made at each step of dynamic programming algorithm and back track to find the optimum sentence alignment.

5. Additionally, we can use heuristics like allowing mapping of multiple sentences in the vicinity of the given sentence to the corresponding sentence in other document, such as to cover cases of splitting a sentence into sentences or vice-versa. But such cases occur rarely and this step can safely be neglected.

4.5 Results

To evaluate our result, we also implemented the Nelken and Shieber(2006)'s approach to compare their results with our results and get a better idea of our method's performance. We chose Nelken's(2006) approach because they have shown that it out performs all other methods. We tested our algorithm on the dataset and found that our approach yielded a precision of 78.84% on a recall of 67.21% giving us an F1-score of 0.7256 . While on the same dataset, Nelken and Shieber's approach gave 65.95% precision on a recall of 50.81% and thus an F1-score of 0.5739. Thus, our approach clearly outperforms Neilken and Shieber's approach. It is to be noted that Nelken and Shieber report an F1-score of 0.6676 at a recall of 0.558, while our implementation of their approach achieved an F1-score of 0.5739 at recall of 0.508. The change in F1-score may be because of the different types of dataset used in the two experiments. Nelken and Shieber had used Britannica encyclopedia and its elementary version containing information about the cities. We have used news reports and their abridged versions which used widely divergent language forms, such as abundant use of change of tense, change of grammatical person, change of writing style etc. which could not be captured by their TF-IDF based similarity. Fig. 1 shows one instance of alignment of a document pair by our approach vs. the gold alignment.

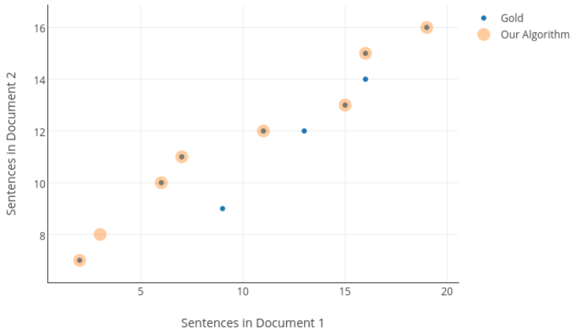
Approach	Precision	Recall	F1-score
RAE+Pool+Align	0.7884	0.6721	0.7256
Nelken's	0.6595	0.5081	0.5739

Table 1: Results of different approaches on dataset

5 Conclusion

We have presented a novel algorithm for aligning the sentences of monolingual corpora of comparable documents. We used a neural network model to arrive at a measure of similarity between sentences. The contextual information present in the document was leveraged upon by using a dynamic programming algorithm to align sentences. Our algorithm performed better than the baseline implementation. It takes into account the semantics being conveyed by the sentences rather just relying on the bag-of-words model for sentence similarity

Figure 1: Gold Assignment vs Our Approach on an example. The orange circles with blue dot denote True Positives, orange circles denote False Positives and the blue dots denote False Negatives.



function.

Acknowledgments

We would like to thank all the anonymous reviewers for their valuable feedback.

References

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pages 25–32.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *journal of machine learning research* 3(Feb):1137–1155.

Paul Clough, Robert Gaizauskas, Scott SL Piao, and Yorick Wilks. 2002. Meter: Measuring text reuse. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pages 152–159.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 160–167.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*. volume 6, pages 449–454.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 350.

Vasileios Hatzivassiloglou, Judith L Klavans, Melissa L Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R McKeown. 2001. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the NAACL workshop on automatic summarization*. volume 1.

Erwin Marsi and Emiel Krahmer. 2005. Explorations in sentence fusion. In *Proceedings of the European Workshop on Natural Language Generation*. Citeseer, pages 109–117.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*. volume 13, pages 746–751.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48(3):443–453.

Rani Nelken and Stuart M Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *EACL*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Advances in Neural Information Processing Systems 24*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, pages 384–394.