Statistical Relational Learning to Recognise Textual Entailment

Miguel Rios[1], Lucia Specia[2], Alexander Gelbukh[3], and Ruslan Mitkov[1]

[1] University of Wolverhampton,
Research Group in Computational Linguistics,
Stafford Street, Wolverhampton, WV1 1SB, UK
{M.Rios,R.Mitkov}@wlv.ac.uk
[2] University of Sheffield,
Department of Computer Science,
Regent Court, 211 Portobello, Sheffield, S1 4DP, UK
L.Specia@sheffield.ac.uk
[3] Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Mexico City, Mexico
www.gelbukh.com

Abstract. We propose a novel approach to recognise textual entailment (RTE) following a two-stage architecture – alignment and decision – where both stages are based on semantic representations. In the alignment stage the entailment candidate pairs are represented and aligned using predicate-argument structures. In the decision stage, a Markov Logic Network (MLN) is learnt using rich relational information from the alignment stage to predict an entailment decision. We evaluate this approach using the RTE Challenge datasets. It achieves the best results for the RTE-3 dataset and shows comparable performance against the state of the art approaches for other datasets.

1 Introduction

Recognising Textual Entailment (RTE) consists in deciding, given two text segments, whether the meaning of one segment (the (H)ypothesis) is entailed from the meaning of the other segment (the (T)ext) [7]. In order to address the task of RTE, most methods rely on machine learning algorithms. For example, a baseline method proposed by Mehdad and Magnini [18] measures the word overlap between the T-H pairs. An overlap threshold is computed over some training data.

Another approach for RTE is to determine some sort of alignment between the T-H pairs. Since T is usually longer, H is aligned to a portion of T, and the best alignment is used to compute a similarity score. A limitation of such approaches is that instead of recognising a non-entailment, an alignment that fits an optimisation criterion will be returned [17], and thus the alignment by itself is a poor predictor for non-entailment. To solve this problem, de Marneffe et al. [17] divide the RTE task such that the alignment and the entailment decision are separate processes. The alignment phase is based on matching graph representations (i.e. dependency relations) of the T-H pair. For the

15th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014, Part I. Lecture Notes in Artificial Intelligence, N 8403, 2013, pp. 330–339. See http://link.springer.com/chapter/10.1007%2F978-3-642-54906-9_27.

entailment decision, rules which strongly suggest implications are designed. A specific rewrite rule between T and H can be positive if they represent entailment or negative otherwise.

Except for Garrette et al. [8], previous work using machine learning is based on propositional representations with simple attribute-value pairs as features. Garrette et al. [8] combines first order logic and statistical methods for RTE. The approach uses discourse structures to represent T-H pairs, and a Markov Logic Network (MLN) model to perform inference in a probabilistic manner over implicativity and factivity, word meaning, and coreference. A threshold on the entailment decision given the MLN model output is manually set. Since their phenomena of interest are not present in the standard RTE datasets, they use handmade datasets. For other related work in the field, we refer the reader to [1].

In this paper we describe an RTE approach following a multi-stage architecture. In contrast to de Marneffe et al. [17], both stages are based on semantic representations in an attempt to measure entailment based on the similarity of answers to the questions *Who did what to whom, when, where, why and how*. This is done through shallow semantic parsing using a Semantic Role Labelling (SRL) tool. Furthermore, instead of using simple similarity metrics to predict the entailment decision, we use rich relational features extracted from the output of the predicate-argument alignment structures between T-H pairs. These are fed to an MLN framework, which learns a model to reward pairs with similar predicates and similar arguments, and penalise pairs otherwise. Different from [8], we do not use a manually set threshold for the entailment decision and we evaluate our method on the standard RTE Challenge datasets, which are larger and contain naturally occurring linguistic constructions that can have an effect on the entailment decision. We compare our approach to previous works for RTE based on alignment techniques, and on probabilistic modelling. Our approach achieves the best performance on the RTE-3 dataset, and competitive results on other datasets.

2 Experimental Design

Our approach to RTE is based on a two-stage architecture: i) alignment, where predicate-argument structures of H and T are aligned; and ii) entailment decision, where the alignments are considered to extract features (i.e. first order logic predicates) and these are used to build an MLN model.

2.1 Alignment Stage

We represent the T-H pair with SRLs as generated by SENNA [6] and use TINE [20, 21] to align any number of predicates and arguments between T and H. Instead of simply matching surface forms, TINE performs a flexible alignment of verb predicates by measuring (i) how similar their arguments are ($argScore$), (ii) and how related the predicates realisations are ($lexScore$). Both scores are combined as shown in Equation 1 to measure the similarity between the two predicates (Av, Bv) from a pair of sentences (A, B).

15th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2014, Part I. Lecture Notes in Artificial Intelligence, N 8403, 2013, pp. 330–339. See http://link.springer.com/chapter/10.1007%2F978-3-642-54906-9_27.

$$\begin{aligned} sim(Av, Bv) = {} & wlex \times lexScore(Av, Bv) \\ & + warg \times argScore(Aarg, Barg) \end{aligned} \tag{1}$$

where $wlex$ and $warg$ are the weights for each component, $argScore(Aarg, Barg)$ is the similarity between the arguments, computed as the cosine distance between the bag-of-words of the predicates' arguments Av, Bv. $lexScore(Av, Bv)$ is the similarity score of the predicates extracted using Dekang Lin's thesaurus [14]. The pair of predicates that maximise Equation 1 produces an alignment with an one-to-one verb-arguments relation.

2.2 Entailment Decision Stage

In the entailment decision stage we use an MLN to predict the entailment relation of a given T-H pair. Statistical relational learning [9], as opposed to a propositional formalism, is focused on representing and reasoning over domains with a relational and probabilistic structure. These models use first-order representations to describe the relations between the domain variables and probabilistic graphical models to reason over uncertainty.

MLN [19] provides a natural choice for this task as it unifies first order logic and probabilistic graphical models in a framework that enables the representation of rich relational information (such as syntactic and semantic relations) and inference under uncertainty. This framework learns weights for first order logic formulas, which are then used to build Markov networks that can be queried in the presence of new instances.

As an inherently semantic task, RTE should naturally benefit from knowledge about the relationships among elements (variables) in a text, in particular to check whether (some of) these relationships are equivalent in both T and H. It is extremely difficult to fully capture relational knowledge using standard propositional formalisms (attribute-value pairs), as it is hard to predict how many elements are involved in a relationship (e.g., a compound argument) or all possible values of these elements, and it is not possible to represent the sharing of values across attributes (e.g. the agent of a predicate which is also the object of another predicate).

The basis for our first order logic formulas are the alignments produced in the previous stage. At inference time, an aligned pair with similar situations and similar participants will likely hold an entailment relation. An alignment consists of a pair of verbs and their corresponding arguments. Several features extracted from these alignments are used as predicates to build a Markov Network. We formulate a relational model based on these predicates along with shallow features used to support the decision when there is no evidence of an alignment for a T-H pair.

Relational Model Our model takes advantage of MLN's ability to handle relational information, and it also takes into consideration the semantic relations between the arguments and verbs. The motivation to design the relational formulas is based on how the alignment stage works. The alignment is performed via heuristics, which means that some of the decisions may introduce incorrect or poor information about the relations between the participants and situations of the entailment candidate pair. In order to alleviate this problem, the relational features reward or penalise each of the aligned verbs from the first stage by making explicit their semantic relation. In addition, the relational features generalise each of the arguments aligned by TINE.

The following variables are created to represent this information: Arg and $Verb$. Figure 1 shows the relationships between these variables in a Markov network.

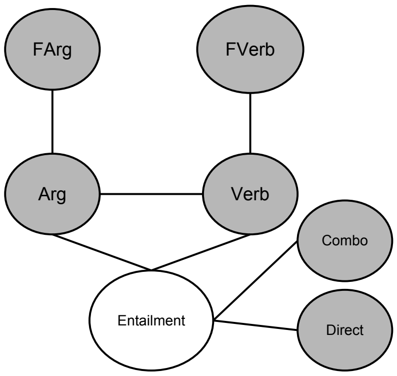

Fig. 1. Markov network of our RTE model

The value of Arg is the label given by the SRL parser for the aligned arguments (e.g., ARG1). The value of $Verb$ is the lexical realisation of the verbs, i.e., the aligned verbs themselves. Furthermore, the aligned arguments and the aligned verbs have features: $FArg$ is the set of features related to the arguments, and $FVerb$ is the set of features related to the verbs.

The features for each token of aligned arguments are as follows:

Lexical Word, lemma and PoS of each token.

Similar Words The 20 most similar words from Dekang Lin's thesaurus for each token. A predicate is created for each similar word.

Hypernyms The first three levels of the hypernym tree above each noun in its first sense in WordNet. A predicate for each hypernym is created.

These argument features are represented by the following formula:

$$Token(aid, pid, +tfeature) \wedge Arg(aid, vid, pid) \Rightarrow Entailment(+d, pid)$$

where $tfeature$ takes the value of each of the previous features, aid and vid are the values of the Arg and $Verb$ variables

For the aligned verbs, the following features are extracted:

Bag-of-words VerbNet $bowfeature$ is the lexical realisation of the classes shared between the verbs in VerbNet. Looking at the semantic classes of the aligned verbs brings extra information about how similar they are:

$$BowVN(vid, +bowfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid)$$

Strong Context $strfeature$ compares components in Equation 1. If the value of $argScore(Aarg, Barg)$ is larger than that of $lexScore(Av, Bv)$, this feature is set to 1, i.e., the similarity of the context of the aligned verbs is stronger than the relationship between them; it is 0 otherwise:

$$StrongCon(vid, +strfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid)$$

Similarity VerbNet $simvnfeature$ is set to 1 if the verbs share at least one class in VerbNet; 0 otherwise:

$$SimVN(vid, +simvnfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid)$$

Similarity VerbOcean $simvofeature$ is 1 if the verbs have the *similar* relation as given by VerbOcean [5];[4] 0 otherwise:

$$SimVO(vid, +simvofeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid)$$

Token Verbs The predicate contains the lemmas of the aligned verbs:

$$TokenVerb(vid, +tokenvfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid)$$

Finally, the relation between Arg and $Verb$ is defined by the formula:

$$Arg(aid, vid, pid) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid)$$

The formulas sharing variables vid and aid indicate relationships between the aligned arguments and the aligned verbs, as well as their corresponding features given the SRL structure. pid relates the previous predicates to the decision of an entailment pair. Many of these formulas can take up multiple values through multiple groundings (e.g. the hypernyms of nouns). The predicate $Entailment(+d, pid)$ takes two possible values for the decision d: $true$ or $false$. The $+$ operator indicates that a weight will be learned for each grounding of the formula. The entailment decision is a hidden variable in the MLN model and it is used to query the MLN.

In the alignment stage, sometimes TINE cannot align a T-H pair, mostly because SENNA does not produce any SRL structure for certain T-H pairs. To be able to make a decision for these pairs using MLNs, we add the variables *Combo* and *Direct* as shallow supporting features for the entailment decision in Figure 1. *Combo* holds the value $cfeature$, which consist of all the combinations of unigrams between the H-T pair. The following predicate is defined for each unigram combination:

$$Combo(pid, +cfeature) \Rightarrow Entailment(+d, pid)$$

[4] VerbOcean contains different relations between verbs.

The *Direct* variable holds the value $simdfeature$ with 1 if the verbs hold an entailment relation as given by the Directional Database [13];[5] 0 otherwise:

$$Direct(pid, +simdfeature) \Rightarrow Entailment(+d, pid)$$

The Markov network built from these formulas can then be queried for an entailment decision. For a new T-H pair, the model predicts a decision based on the type of arguments it has, the features of the words in the arguments, the alignment between its verbs, the relations between such verbs, and the shallow support features.

3 Experimental Results

We use the Alchemy[6] toolkit and the datasets from the RTE challenges 1-3 [7, 2, 10], which are publicly available, to evaluate our MLN model. To predict the entailment decision we take the marginal probabilities that Alchemy outputs for a given query, i.e., the $Entailment$ predicate. The query with the highest probability gives the entailment decision.

For a fair comparison, we evaluate our approach against previous work for RTE that is also based on alignment techniques. de Marneffe et al. [17] use a two-stage alignment similar to ours, but with dependency trees instead of SRLs. In addition, the entailment decision problem is represented with a vector of 54 features, where these features try to capture entailment and non-entailment by focusing on negations and quantifiers. Training and is performed using a logistic regression classifier. Chambers et al. [4] improve the alignment stage in [17] and combine it with a logical framework for the second stage [16]. The inference in the logical framework is expressed by a sequence of edits over texts expressions, where the edits represent operations that affect monotonicity over texts expressions. The logical framework maps alignments into a sequence of edits that defines the entailment decision. MacCartney et al. [15] propose a phrase-base alignment that uses external lexical resources. They improve the first stage via knowledge about semantic similarity and an extra, specific dataset for the training of the alignment stage.

Table 1. Accuracy against previous work based on alignment over the RTE datasets

Method	RTE-1	RTE-2	RTE-3
de Marneffe et al. [17]	-	60.5%	60.5%
Chambers et al. [4]	-	-	63.62%
MacCartney et al. [15]	-	60.3%	-
Relational Model	57%	55%	65%

Table 1 shows that our approach outperforms previous work for the RTE-3 dataset. However, the results are less positive for RTE-2. A possible reason for this error is the

[5] It contains directional lexical entailment rules.

[6] `http://alchemy.cs.washington.edu/`

low performance of our alignment technique. TINE only finds alignments for a subset of the test sets: 162 pairs (out of 287) for RTE-1, 463 pairs (out of 800) for RTE-2, and 385 pairs (out of 800) for RTE-3. Therefore, the proportionally fewer noisy alignments obtained for RTE-3 could have contributed to the better performance of the approach on this dataset. Another reason for the differences in performance across datasets can be the way the RTE datasets were built. RTE-3 contains longer T parts, with longer contexts, and therefore our method can find good quality alignments. This also seem to affect the overall performance of the participating systems, since the average accuracy (across all participating systems) for RTE-1 is 55%, while it is 59% for RTE-2, and 61% for RTE-3.

Our approach predicts a larger proportion of the *TRUE* class for RTE-3 than for RTE-2. There is a big gap between precision (54%) and recall (70%) for the RTE-3 dataset. Whereas for the RTE-2 this gap is smaller, with 52% precision and 57% recall. This behaviour could be because TINE finds more alignments for the *TRUE* pairs.

To further analyse the impact of poor alignment decisions, we test our model on the subsets of the datasets for which TINE produced an alignment. We compare the relational model only with the alignment features (i.e. without the shallow features) against a Support Vector Machine (SVM)-based approach. For the SVM algorithm, we compute a common and strong RTE baseline: the overlap of lemmas between T-H pairs as features, and use a linear kernel to learn the binary entailment decision [18]. Table 2 shows the results, where the relational model clearly outperforms the SVM model, and by a large margin on the RTE-3 dataset. This shows the potential of the relational features and MLNs for RTE.

Table 2. Accuracy on a subset of RTE 1-3 where an alignment is produced by TINE for T-H

Algorithm	RTE-1	RTE-2	RTE-3
SVM	50%	51%	56%
Relational model	57%	55%	78%

For a comparison covering the other main aspect of our approach – its probabilistic nature –, in a second evaluation experiment we compare our approach against other methods based on probabilistic modelling.

Glickman and Dagan [11] model entailment via lexical alignment, where the web co-occurrences for a pair of words are used to describe the probability of the hypothesis given the text. Harmeling [12] propose a model that, with a given sequence of transformations over a parse tree, keeps entailment decisions with a certain probability. Wang and Manning [22] merge the alignment and the decision into one step, where the alignment is a latent variable. The alignment is used into a probabilistic model that learns tree-edit operations on dependency parse trees. Beltagy et al. [3] extend the work in [8] to be able to process large scale datasets such as those from the RTE challenges. The method transforms distributional similarity judgments to weighted inference formulas, where the distributional similarity (i.e. If X and Y occur in similar contexts they describe similar entities) describes the degree of entailment between pairs.

Table 3. Accuracy against previous work based on probabilistic modelling over the RTE datasets

Method	RTE-1	RTE-2	RTE-3
Glickman and Dagan [11]	59%	-	-
Harmeling [12]	-	-	59.3%
Wang and Manning [22]	-	63%	61.1%
Beltagy et al. [3]	57%	-	-
Relational Model	57%	55%	65%

Table 3 shows a similar behaviour as the previous comparison: our approach leads to considerably better results on RTE-3, but lower performance for RTE-2. In addition, for the RTE-1 dataset, which has also been used by most of these other approaches, our relational model shows very competitive performance. In particular, it achieves the same performance as Beltagy et al. [3], which also use a MLN for the entailment decision.

4 Conclusions

We have described a proposal on using a relational statistical learning framework for the RTE task. Our experiments showed promising results. The main source of errors was found to be the alignment step, which has low coverage and can produce noisy alignments. However, we showed that when an alignment is found, the relational features improve the final entailment decision.

Future work includes improvements in the alignment stage as well as incorporating a more robust set of support features, such as using syntactic structures along with the semantic structures into a combined relational model. In other words, we could use different types of alignments (e.g., monolingual word alignment, syntactic alignment) that are based on heuristics, where the objective of the MLN formulas will be to penalise or reward the decisions made by different aligners. We also plan to define formulas that relate decisions across aligners.

Acknowledgments

This work was supported by the Mexican National Council for Science and Technology (CONACYT), scholarship reference 309261, by the QTLaunchPad (EU FP7 CSA No. 296347) project, and Governments of Mexico and India (DST-CONACYT grant 122030; SIP IPN grant 20144534; COFAA-IPN).

References

[1] Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. J. Artif. Int. Res. 38(1), 135–187 (2010)

[2] Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The second pascal recognising textual entailment challenge. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment. Venice, Italy (2006)

[3] Beltagy, I., Chau, C., Boleda, G., Garrette, D., Erk, K., Mooney, R.: Montague meets markov: Deep semantics with probabilistic logical form. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pp. 11–21. Atlanta, Georgia, USA (June 2013)

[4] Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M.C., Ramage, D., Yeh, E., Manning, C.D.: Learning alignments and leveraging natural logic. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 165–170. Association for Computational Linguistics, Prague (June 2007)

[5] Chklovski, T., Pantel, P.: Verbocean: Mining the web for fine-grained semantic verb relations. In: Proceedings of EMNLP 2004. pp. 33–40 (2004)

[6] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537 (2011)

[7] Dagan, I., Glickman, O.: The pascal recognising textual entailment challenge. In: Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment (2005)

[8] Garrette, D., Erk, K., Mooney, R.: Integrating logical representations with probabilistic information using Markov logic. In: Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011). pp. 105–114 (2011)

[9] Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)

[10] Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third pascal recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 1–9. Prague (2007)

[11] Glickman, O., Dagan, I.: M.: A lexical alignment model for probabilistic textual entailment. this volums. In: Lecture Notes in Computer Science. pp. 287–298. Springer (2006)

[12] Harmeling, S.: An extensible probabilistic transformation-based approach to the third recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 137–142. Association for Computational Linguistics, Prague (June 2007)

[13] Kotlerman, L., Dagan, I., Szpektor, I., Zhitomirsky-geffet, M.: Directional distributional similarity for lexical inference. Nat. Lang. Eng. 16(4), 359–389 (2010)

[14] Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics. pp. 768–774. Montréal, Canada (1998)

[15] MacCartney, B., Galley, M., Manning, C.D.: A phrase-based alignment model for natural language inference. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. pp. 802–811. Association for Computational Linguistics, Honolulu, Hawaii (October 2008)

[16] MacCartney, B., Manning, C.D.: Natural logic for textual inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 193–200. Association for Computational Linguistics, Prague (June 2007)

[17] de Marneffe, M.C., MacCartney, B., Grenager, T., Cer, D., Rafferty, A., Manning, C.D.: Learning to distinguish valid textual entailments. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment. Venice, Italy (2006)

[18] Mehdad, Y., Magnini, B.: A word overlap baseline for the recognizing textual entailment task. Available: `http://hlt.fbk.eu/sites/hlt.fbk.eu/files/baseline.pdf` (2009)

[19] Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)

[20] Rios, M., Aziz, W., Specia, L.: TINE: A metric to assess MT adequacy. In: Proceedings of the Sixth Workshop on Statistical Machine Translation. pp. 116–122. Edinburgh, Scotland (2011)

[21] Rios, M., Aziz, W., Specia, L.: UOW: Semantically informed text similarity. In: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). pp. 673–678. Montréal, Canada (2012)

[22] Wang, M., Manning, C.D.: Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. In: Proceedings of the 23rd International Conference on Computational Linguistics. pp. 1164–1172. COLING '10, Stroudsburg, PA, USA (2010)