Recognizing Textual Entailment with Similarity Metrics

Miguel Rios[1] and Alexander Gelbukh[2]

[1] University of Wolverhampton,
Research Group in Computational Linguistics,
Stafford Street, Wolverhampton, WV1 1SB, UK
M.Rios@wlv.ac.uk
[2] Center for Computing Research,
National Polytechnic Institute,
Mexico City, Mexico
www.gelbukh.com

Abstract. We present a system for the Recognizing Textual Entailment task, based on various similarity metrics, namely (i) string-based metrics, (ii) chunk-based metric, (iii) named entities-based metric, and (iv) shallow semantic metric. We propose the chunk-based and named entities-based metrics to address limitations of previous syntactic and semantic-based metrics. We add the scores of the metrics as features for a machine learning algorithm. Then, we compare our results with related work. The performance of our system is comparable with the average performance of the Recognizing Textual Entailment challenges systems, though lower than that of the best existing methods. However, unlike more sophisticated methods, our method uses only a small number of simple features.

1 Introduction

The Recognizing Textual Entailment (RTE) task consists in deciding, given two textual expressions, whether the meaning of one of them, called Hypothesis (H), is entailed by the meaning of the other one, called Text (T) [5]. The RTE Challenge is a generic task which addresses common semantic inference needs across Natural Language Processing (NLP) applications.

In order to address the task of RTE, different methods have been proposed. Most of these methods rely on machine learning (ML) algorithms. For example, a baseline method proposed by Mehdad and Magnini [9] consists in measuring the word overlap between the Text and Hypothesis; the word overlap is the number of words shared between the two textual expressions. Their method is organized into three main steps: (i) pre-processing: all T–H pairs are tokenized and lemmatized; (ii) computing of the word overlap; (iii) building a binary classifier. An overlap threshold is computed over the training data, and the test data is classified based on the learned threshold. If the word overlap score is greater than the threshold, then the entailment decision is TRUE (there is entailment), otherwise

Advances in Computer Science and Engineering
Research in Computing Science 58, 2012 pp. 339–349
Paper Recived and Acepted UK, Mexico

it is FALSE (there is no entailment). The motivation behind this paradigm is that a T–H pair with a strong similarity score has good chances to represent an entailment relation. Different types of similarity metrics are applied over the T–H pair in order to extract features and to train a classifier.

Similarity metrics that deal with semantics usually use information from ontologies or semantic representations given by parsers [2]. However, the comparison between texts is done by matching the semantic labels, and not by matching the content of those units.

In this work we describe an RTE system based on various similarity metrics. In addition, we propose new similarity metrics based on different representations of text for RTE that are: (i) chunks and (ii) Named Entities. The goal of the introduction of these new features is to address limitations of previous syntactic- and semantic-based metrics. We add the scores of the new metrics along with simple string-based similarity metrics and a shallow-semantic-based metric [11] as features for a machine learning method for RTE. Then, we compare our results with related work on RTE. The performance of our system is comparable with the average performance of the RTE challenges, though it is lower than that of the best known methods.

In the remainder of this paper we discuss the related work (Section 2), describe our RTE system (Section 3) and compare its performance with previous work (Section 4). Finally, we give conclusions and suggest some future work (Section 5).

2 Related Work

Burchardt et al. [2] introduced new features for RTE that involve deep linguistic analysis and shallow word overlap measure. Their method consists of three steps: first, they represent the T–H pair with the Frame Semantics (FS) and Lexical Functional Grammars (LFG) formalisms; this representation is similar to the Semantic Role Labeling. Then, they calculate a similarity score based on matching the LFG graphs, and finally make a statistical entailment decision. They used the RTE-2 and RTE-3 datasets as training data, and extracted 47 features from the deep and the shallow overlap. These features consist of combinations of predicates overlaps, grammatical functions match, and lexical overlaps.

The methods that use Semantic Role Labeling (SRL) for RTE use the annotation provided by a semantic parser to measure the similarity between texts. However, they only measure the similarity in terms of how many labels the two texts share (overlaps) and not in termos of the content marked with those labels.

Delmonte et al. [8] introduced semantic-mismatch features, such as locations, discourse markers, quantifiers, and antonyms. Their entailment decisions are based on applying rewards and penalties over the semantic similarity and shallow similarity scores. Later, Delmonte et al. [6] participated in the RTE-2 challenge with an enhanced version of their previous system. Their new system uses new features based on heuristics, such as Augmented Head Dependency Structures, grammatical relations, negations, and modal verbs.

Roth and Sammons [12] used semantic logical inferences for RTE, where the representation method is a Bag-of-Lexical-Items (BoLI). The BoLI relies in word overlap. It states that the entailment relation holds if the overlap score is above a certain threshold. An extended set of stopwords is used to select the most important concepts for the BoLI, such as auxiliary verbs, articles, exclamations, discourse markers, and words in WordNet. Also, in order to recognize relations in the T–H pairs, the system checks matchings between SRLs, and then applies a series of transformations over the semantic representations to make easier to determine the entailment. Their system uses the following transformation operations:

- *annotate*, which make some implicit property of the meaning of the sentence explicit;
- *simplify* and *transform*, which remove or alter some section of the text T in order to improve annotation accuracy or make it more similar to H;
- *compare*, which compares some elements of the two members of the entailment pair and assigns a score that correlates to how successfully those elements of the H can be subsumed by the T.

3 Experimental Design

The RTE task can be seen as a binary classification task, where the entailment relations are the classes. Then the RTE benchmark datasets can be used to train a classifier [4].

Our RTE system is based on a supervised machine learning algorithm. We train the machine learning algorithm with similarity scores computed over the T–H pairs extracted from different classes of metrics described below.

With these metrics we build a vector of similarity scores used as features to train a machine learning algorithm. We use the development datasets from the RTE 1 to 3 benchmark to train different ML algorithms, using their implementations from the WEKA toolset[3] without any parameter optimization. Then, we test the models with a tenfold cross-validation over the development datasets to decide which algorithm to use for the comparison against related work over the test datasets.

The metrics we used as as follows.

3.1 Lexical Metrics

We use the following string-based similarity metrics: precision, recall, and F_1:

$$\text{precision}(T, H) = \frac{|T \bigcap H|}{|H|} \tag{1}$$

$$\text{recall}(T, H) = \frac{|T \bigcap H|}{|T|} \tag{2}$$

[3] http://www.cs.waikato.ac.nz/ml/weka/

$$F_1(T,H) = 2 \times \frac{\text{precision}(T,H) \times \text{recall}(T,H)}{\text{precision}(T,H) + \text{recall}(T,H)} \tag{3}$$

As input for the metrics we use a bag-of-words (BoW) representation of the T–H pairs. However, we only use content words to compute the similarity score in the T–H pairs.

3.2 Chunking

Shallow parsing (or chunking) consists in tagging a text with syntactically correlated parts. This alternative to full parsing is more efficient and more robust. Chunks are non-overlapping regions of text that are sequences of constituents that form a group with a grammatical role (e.g. noun group). The motivation for introducing a chunking similarity metric consists in that a T–H pair with a similar syntax structure can hold an entailment relation. The chunking feature is defined as the average of the number of similar chunks (in the same order) in the T–H pair:

$$\text{chunking}(T,H) = \frac{1}{m} \sum_{n=1}^{m} simChunk(t_n, h_n), \tag{4}$$

where m is the number of chunks in T, t_n is the n-th chunk tag and content in the same order, and $simChunk(t_n, h_n) = 1$ if the content and annotation of the chunk are the same, and 0.5 if the content of the chunk is different but the chunk tag is still the same.

The following example shows how the chunking metric works. Consider:

T: *Along with chipmaker Intel , the companies include Sony Corp. , Microsoft Corp. , NNP Co. , IBM Corp. , Gateway Inc. and Nokia Corp.*
H: *Along with chip maker Intel , the companies include Sony , Microsoft , NNP , International Business Machines , Gateway , Nokia and others.*

First, for each chunk, this metric compares and scores the content of the tag: whether it is the same chunk group and whether it is the same order of chunks. Table 1 shows how this metric scores each chunk for the previous example.

Finally, the chunking metric (4) computes the individual scores and gives a final score of $chunking(T,H) = 0.64$ for this example.

3.3 Named Entities

Named Entity Recognition (NER) is a task that identifies and classifies parts of a text into predefined classes such as *names of persons*, *organizations*, *locations*, *expressions of times*, *quantities*, *monetary values*, *percentages*, etc. For example, from the text: "Acme Corp bought a new...", *Acme Corp* is identified as a named entity and classified as an *organization*.

Table 1. Example of partial scores given by the chunking metric

Tag	Content	Tag	Content	Score
PP	*Along*	PP	*Along*	1
PP	*with*	PP	*with*	1
NP	*chipmaker Intel*	NP	*chip maker Intel*	0.5
NP	*the companies*	NP	*the companies*	1
VP	*include*	VP	*include*	1
NP	*Sony Corp.*	NP	*Sony*	0.5
NP	*Microsoft Corp.*	NP	*Microsoft*	0.5
NP	*IBM Corp.*	NP	*International Business Machines*	0.5
NP	*Gateway Inc.*	NP	*Gateway*	0.5
NP	*Nokia Corp.*	NP	*Nokia and others.*	0.5

The motivation of a similarity measure based on NER is that the participants in H should be the same as those in T, and H should not include more participants in order to hold an entailment relation. The goal of the measure is to deal with synonymous entities.

Our approach for the NER similarity measure consists in the following: first, the named entities are grouped by type; then, the content of the same type of groups (e.g *Scripps Hospital* is an organization) is compared using the cosine similarity equation. However, if the surface realizations of the same named entity in T and H are different, we retrieve words that share the same context as the named entity in question; the words are retrieved from Dekang Lin's thesaurus. Therefore, the cosine similarity equation will have more information than just the named entity.

For instance, consider the T–H pair from the previous example. The entity from T: *IBM Corp.* and the entity from H: *International Business Machines* have the same tag *organization*. Our metric groups them and adds words from the similarity thesaurus resulting in the following bag-of-words (BoW) representation:

T's entities: {*IBM Corp., ..., Microsoft, Intel, Sun Microsystems, Motorola / Motorola, Hewlett-Packard / Hewlett-Packard, Novell, Apple Computer, ...*}

H's entities: {*International Business Machines, ..., Apple Computer, Yahoo, Microsoft, Alcoa, ...*}.

Finally, the metric computes the cosine similarity between these BoWs.

3.4 TINE

TINE [11] is an automatic metric based on the use of shallow semantics to align predicates and their respective arguments between a pair of sentences. The metric combines a lexical matching with a shallow semantic component to address adequacy for machine translation evaluation. The goal of this metric is to provide a flexible way of align shallow semantic representations (semantic role

labels) by using both the semantic structure of the sentence and the content of the semantic components.

A verb in the hypothesis H is aligned to a verb in the text T if they are related according to the following heuristics: (i) the two verbs share at least one class in VerbNet, or (ii) the pair of verbs holds a relation in VerbOcean.

For example, in VerbNet the verbs *spook* and *terrify* share the same class, namely, *amuse-31.1*, and in VerbOcean the verb *dress* is related to the verb *wear*.

The following example shows how the alignment of verbs and predicates is performed. Consider:

H: *The lack of snow discourages people from ordering ski stays in hotels and boarding houses.*
T: *The lack of snow is putting people off booking ski holidays in hotels and guest houses.*

Then, the algorithm proceeds with the following steps:

1. Extract verbs from H: $V_H = \{discourages, ordering\}$
2. Extract verbs from T: $V_T = \{putting, booking\}$
3. Similar verbs aligned with VerbNet (shared class get-13.5.1):
   V = $\{(v_H = order, v_T = book)\}$
4. Compare arguments of ($v_H = order$, $v_T = book$):
   A_H = {A0, A1, AM-LOC}
   A_T = {A0, A1, AM-LOC}
5. $A_H \cap A_T$ = {A0, A1, AM-LOC}
6. Exact matches:
   $H_{A0} = \{people\}$ and $T_{A0} = \{people\}$
7. Different word forms:
   expand the representation:
   $H_{A1} = \{ski, stays\}$ and $T_{A1} = \{ski, holidays\}$
   to:
   $H_{A1} = \{\{ski\}, \{stays, remain, ..., journey, ...\}\}$
   $T_{A1} = \{\{ski\}, \{holidays, vacations, trips, ..., journey, ...\}\}$
8. Similarly with H_{AM-LOC} and T_{AM-LOC}

Here, V_H is the set of verbs in the hypothesis H, V_T is the set of verbs in the text T, A_H is the set of arguments of the hypothesis H, and A_T is the set of arguments in the text T.

The metric aligns similar verbs with the ontology and similar arguments with a distributional thesaurus. Then, the metric computes a similarity score given the previous alignment points.

4 Experimental Results

We compared our method with other machine learning-based methods and with methods that use a SRL representation as one of its features.

Table 2. The 10-fold cross-validation accuracy results over the RTE development datasets

Algorithm	RTE-1	RTE-2	RTE-3
SVM	**64.90%**	**59.00%**	**66.62%**
NaïveBayes	62.25%	58.25%	64.50%
AdaBoost	**64.90%**	57.75%	62.75%
BayesNet	64.19%	**59.00%**	65.25%
LogitBoost	62.25%	52.50%	61.00%
MultiBoostAB	64.55%	**60.50%**	64.00%
RBFNetwork	61.90%	54.25%	64.80%
VotedPerceptron	63.31%	57.75%	65.80%

We used the RTE-1, RTE-2, and RTE-3 development datasets to train the classifiers. Table 2 shows the tenfold cross-validation results.

The SVM achieved the best results in the experiments during the training phase. We use this algorithm to perform the classification over the RTE test datasets. The data used for classification are the test datasets of the RTE challenge. The experimental results are summarized in Table 3.

Table 3. Comparison with previous accuracy results over the RTE test datasets

Method	RTE-1	RTE-2	RTE-3
Roth and Sammons [12]	–	–	65.56%
Burchardt and Frank [1], Burchardt et al. [2]	54.6%	59.8%	62.62%
Delmonte et al. [8], [6], [7]	59.25%	54.75%	58.75%
Our method with SVM	53.87%	55.37%	61.75%

Table 4 shows the overall accuracy results of the RTE systems on the RTE test datasets against our method. Our method is close to the average performance but below the best method.

However, the systems that showed the best results in the RTE challenge are complex and sophisticated systems. In contrast, our method relies on a small number of simple features. Our main semantic feature is focused in predicate-argument information, while other methods tackle several semantic phenomena such as negation and discourse information [12] or rely on a large number of features [2].

Table 4. Comparison with overall accuracy results over the RTE test datasets

Challenge	Our method	Average	Best
RTE-1	53.87%	55.12%	70.00%
RTE-2	55.37%	58.62%	75.38%
RTE-3	61.75%	61.14%	80.00%

Error analysis shows that the most common source of errors for our method is the TINE similarity metric. The following categories of errors made by this metric are the most common ones:

1. Lack of coverage in the ontologies, for example:

   T: *This year, women were awarded the Nobel Prize in all fields except physics.*
   H: *This year the women received the Nobel prizes in all categories less physical.*

   The lack of coverage in the VerbNet ontology prevented the detection of the similarity between *receive* and *award*.
2. Matching of unrelated verbs, for example:

   T: *If snow falls on the slopes this week, Christmas will sell out too, says Schiefert.*
   H: *If the roads remain snowfall during the week, the dates of Christmas will dry up, said Schiefert.*

   In VerbOcean, *remain* and *say* are incorrectly indicated to be related. The VerbOcean dictionary was created by a semi-automatic extraction algorithm [3] with an average accuracy of 65.5% and thus contain a considerable number of errors.
3. Incorrect tagging of the semantic roles by the semantic parser SENNA[4], for example:

   T: *Colder weather is forecast for Thursday, so if anything falls, it should be snow.*
   H: *On Thursday, must fall temperatures and, if there is rain, in the mountains should.*

   The position of the predicates affects the SRL tagging. The predicate *fall* has the roles (A1, V, and S-A1) in the reference, and the roles (AM-ADV, A0, AM-MOD, and AM-DIS) in the hypothesis H. As a consequence, the metric cannot match the fillers. Also, SRL systems do not detect phrasal verbs: e.g., the action *putting people off* is similar to *discourages* but current SRL systems do not detect this.

As we see, the quality of the semantic parser and the coverage of the ontologies are significant causes that affect the performance of our method.

In addition, on the RTE-1 test dataset with 800 T–H pairs, the coverage of the semantic metric is 491 pairs. This means that the system only predicts a certain amount of pairs. On the RTE-3 dataset, on which we obtain the best result, also has 800 T–H pairs, but the coverage on this dataset is much better: 556 pairs. Accordingly, our method has a smaller amount of errors due to a greater number of semantic-scored pairs.

[4] SENNA, http://ml.nec-labs.com/senna/

5 Conclusions and Future Work

We have presented a machine learning-based system for Recognizing Textual Entailment (RTE) task, based on new similarity metrics as well as simple string-based metrics and a shallow-semantic metric. The new similarity measures are based on: (i) chunking, (ii) named entities.

Our method has performance comparable with the average performance of methods in the RTE challenges. However, its performance is below that of the best know methods. On the other hand, our method relies on a small number of simple features, and our system only tackles one semantic phenomenon: predicate-argument information.

A preliminary error analysis shows that a main source of errors is the alignment of predicates by the TINE measure. However, if the system has more pairs tagged with predicate-argument information, then its performance improves.

In order to improve the performance of our current machine learning-based system, in our future work we will attempt to resolve the errors caused by the TINE metric based on the error analysis, or will use a different semantic approach to RTE [10].

Our semantic metric uses a distributional thesaurus to measure the similarity between arguments, so that, for example, *cat* and *dog* will be aligned because they share the same context. A possible direction to improve the semantic metric is to add hard constraints over the core arguments. These constrains can be defined as thresholds learned over the training dataset.

Acknowlegments

This work was partially supported by the Mexican National Council for Science and Technology (CONACYT), scholarship reference 309261, and SIP-IPN grant 20121823.

References

[1] Burchardt, A., Frank, A.: Approaching textual entailment with LFG and FrameNet frames. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment. Venice, Italy (2006)

[2] Burchardt, A., Reiter, N., Thater, S., Frank, A.: A semantic approach to textual entailment: System evaluation and task analysis. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 10–15. Association for Computational Linguistics, Prague (June 2007)

[3] Chklovski, T., Pantel, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In: Lin, D., Wu, D. (eds.) Proceedings of EMNLP 2004. pp. 33–40. Barcelona, Spain (Jul 2004)

[4] Dagan, I., Dolan, B., Magnini, B., Roth, D.: Recognizing textual entailment: Rational, evaluation and approaches – erratum. Natural Language Engineering 16(1), 105 (2010)

[5] Dagan, I., Glickman, O.: The PASCAL Recognising Textual Entailment challenge. In: In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment (2005)
[6] Delmonte, R., Bristot, A., Boniforti, M.A.P., Tonelli, S.: Coping with semantic uncertainty with VENSES. In: Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment. Venice, Italy (2006)
[7] Delmonte, R., Bristot, A., Piccolino Boniforti, M.A., Tonelli, S.: Entailment and anaphora resolution in RTE 3. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 48–53. Association for Computational Linguistics, Prague (June 2007)
[8] Delmonte, R., Tonelli, S., Piccolino Boniforti, M.A., Bristot, A., Pianta, E.: VENSES – a linguistically-based system for semantic evaluation. In: In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment (2005)
[9] Mehdad, Y., Magnini, B.: A word overlap baseline for the Recognizing Textual Entailment task (2009)
[10] Pakray, P., Barman, U., Bandyopadhyay, S., Gelbukh, A.: A statistics-based semantic textual entailment system. Lecture Notes in Artificial Intelligence 7094, 267–276 (2011)
[11] 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. Association for Computational Linguistics, Edinburgh, Scotland (July 2011)
[12] Roth, D., Sammons, M.: Semantic and logical inference model for textual entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 107–112. Association for Computational Linguistics, Prague (June 2007)