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
gelbukh@gelbukh.com

Abstract. We present a Recognizing Textual Entailment system based on different similarity metrics. The metrics are: i)String-based metrics, ii) Chunks, ii) Named Entities, and iii) Shallow-semantic metric. We propose the Chunks and Named Entities metrics to address limitations of previous syntactic and semantic based metrics. We add the scores of the metrics as features into 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, but the performance is lower with both the related work and the best methods.

1 Introduction

The Recognizing Textual Entailment (RTE) task consists in deciding, given two text expressions, whether the meaning of one text is entailed from the meaning of the other text [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 and 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(T-H) pairs, where the word overlap is the number of words shared between text and hypothesis. The method is divided into three main steps: i) pre-process: All T-H pairs are tokenized and lemmatized. ii) compute the word overlap. iii) build 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 the entailment decision is TRUE, otherwise is FALSE. The motivation behind this paradigm is that a pair with a strong similarity score holds an entailment relation. Then, different types of similarity metrics are

applied over the T-H pair to extract features and 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 different 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 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 metric [11] as features into a ML 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, but the performance is lower with both the related work and the best methods.

In the remainder of this paper we show the related work (Section 2), we describe our RTE system (Section 3) and its performance compared to previous work (Section 4). We then provide conclusions and future work (Section 5).

2 Related Work

Burchardt et al. [2] introduce new features for RTE. The new features as well as other methods involve deep linguistic analysis and shallow word overlap. The method consists of three steps: first, represent the T-H pair with the Frame Semantics (FS) and Lexical Functional Grammars (LFG) formalisms (the representation is similar to Semantic Role Labeling). Second, extract a similarity score based on matching the LFG graphs, and then make a statistical entailment decision. Burchardt et al. [2] use the RTE-2 and RTE-3 datasets as training data, and 47 features are extracted form the deep and the shallow overlap. The features consist of combinations of: predicates overlaps, grammatical functions match and lexical overlaps. The methods which use Semantic Role Labeling (SRL) for RTE use the annotation provided by a semantic parser to measure the similarity between texts, but only measure the similarity in terms of how many labels they share (overlaps) and not the content of those labels.

Delmonte et al. [8] introduce semantic-mismatch features such as: locations, discourse markers, quantifiers and antonyms. The entailment decision is based on applying rewards and penalties over the semantic-similarity and shallow scores. Delmonte et al. [6] participated in the RTE-2 Challenge with an enhanced version of their previous system. The new system consists in new features based on heuristics such as: Augmented Head Dependency Structures, grammatical relations, negations and modal verbs.

Roth and Sammons [12] use semantic logical inferences for RTE, where the representation method is a Bag-of-Lexical-Items (BoLI). The BoLI relies in word overlap, in which an entailment relation holds if the overlap score is above a certain threshold. An extended set of stop words is used to select the most important

concepts for the BoLI (auxiliary verbs, articles, exclamations, discourse markers and words in WordNet). Also, in order to recognize relations over the T-H pairs the system checks matchings between SRL's, and then applies a series of transformations over the semantic representations to make easier to determine the entailment. The transformation operations are: *annotate* make some implicit property of the meaning of the sentence explicit. *Simplify/Transform* remove or alter some section of T in order to improve annotation accuracy or make it more similar to H. *Compare* (some elements of) the two members of the entailment pair and assign a score that correlates to how successfully (those elements of) the H's can be subsumed by T.

3 Experimental Design

The RTE task can be seen as a binary classification task where the entailment relations are the classes, and the RTE benchmark datasets are 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 such as:

Lexical Metrics We use the following string-based similarity metrics: Precision (1), Recall (2) and F-1 (3). We use as input for the metrics a representation of Bag-of-Words (BoW) of the T-H pairs. However, we only use content words to compute the similarity score between the T-H pairs.

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

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

$$F(T,H) = 2 \cdot \frac{precision(T,H) \cdot recall(T,H)}{precision(T,H) + recall(T,H)} \tag{3}$$

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

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

where m is the number of chunks in T, t_n is the n 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 $simChunk(t_n, h_n) = 0.5$ 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:
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 the metric compares and scores the content of the tag if it is the same chunk group and if it is the same order of chunks. Table 1 shows how the metric scores each chunk for the previous example.

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

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

Named Entities Named Entity Recognition (NER) is a task part of Information Extraction which 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.
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 synonym entities.
Our approach for the NER similarity measure consists in the following: First, the Named entities are grouped by type, and then the content of the same type of groups (e.g Scripps Hospital is an Organization) is compared with the cosine similarity equation. But if the surface realization is different we retrieve words that share the same context as the Named Entity (the words are retrieved from the Dekang Lin's thesaurus). Therefore, the cosine similarity equation will have more information than just the Named Entity. For example, from the T-H pair:

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.*
The entity from T: *IBM Corp.* and the entity from H: *International Business Machines* have the same tag *Organization*. The metric groups them and adds words from the similarity thesaurus resulting in the following Bag-of-Words (BoW).
T entity: {*IBM Corp.,... Microsoft, Intel, Sun Microsystems, Motorola/Motorola, Hewlett-Packard/Hewlett-Packard, Novell, Apple Computer...*}
and H entity: {*International Business Machines,... Apple Computer, Yahoo, Microsoft, Alcoa...*}.
Then the metric computes the cosine between the new pair of BoW's.

TINE The 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 is aligned to a verb in the text if they are related according to the following heuristics: (i) the pair of 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 *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:

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.*

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}
   expand to:
   H_{A1} = {{ski},{stays, remain... journey...}}

T_{A1} = {{ski},{holidays, vacations, trips... journey...}}

8. similarly to H_{AM-LOC} and T_{AM-LOC}

Where V_h is the set of verbs in the hypothesis, V_t is the set of verbs in the text, A_h is the set of arguments of the hypothesis and A_t is the set of arguments in the text. 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.

With the previous 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 implementations from WEKA[3] without any parameter optimization. Then, we test the models with a 10-fold cross-validation over the development datasets to decide which algorithm use for the comparison against related work over the test datasets.

4 Experimental Results

We compare our method with ML-based methods, and with methods that use a SRL representation as one of its features. We use the RTE-1, RTE-2, and RTE-3 development datasets to train the classifiers. Table 2 shows the 10-fold cross-validation results.

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%**	**66.62%**
NaïveBayes	62.25%	58.25%	64.50%
AdaBoost	**64.90%**	57.75%	62.75%
BayesNet	64.19%	**59%**	65.25%
LogitBoost	62.25%	52.5%	61%
MultiBoostAB	64.55%	**60.5%**	64%
RBFNetwork	61.90%	54.25%	64.8%
VotedPerceptron	63.31%	57.75%	65.8%

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 4 shows the overall accuracy results of the RTE test datasets against our method. Our method is close to the average performance but far from the

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

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], Delmonte et al. [6], Delmonte et al. [7]	59.25%	54.75%	58.75%
Our method with SVM	53.87%	55.37%	61.75%

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%

best method. However, the related work are complex systems. In contrast, our method relies in less and simple features. Our main semantic feature is focused in predicate-argument information, where other methods tackle several semantic phenomena such as negation and discourse information [12]. Or methods with a large number of features [2].

We discuss with a few examples some of the common errors made by the TINE similarity metric. Overall, we consider the following categories of errors:

1. Lack of coverage of the ontologies.

   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.

   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 said to be related. VerbOcean was created by a semi-automatic extraction algorithm [3] with an average accuracy of 65.5%.
3. Incorrect tagging of the semantic roles by the semantic parser SENNA[4].

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

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 following roles (A1, V, and S-A1) in the reference, and the following roles (AM-ADV, A0, AM-MOD, and AM-DIS) in the hypothesis. As a consequence, the metric cannot attempt to match the fillers. Also, SRL systems do not detect phrasal verbs, where the action *putting people off* is similar to *discourages*.

Above we show with examples that the quality of the semantic parser and the coverage of the ontologies can be reasons which affect the performance of this method. In addition, in the RTE-1 test dataset with 800 T-H pairs the coverage of the semantic metric is 491 pairs. Which means that the system only predicts a certain amount of pairs. In the RTE-3 dataset, which is the model with the best result, with 800 T-H pairs. The coverage for this dataset increases to 556 pairs. Thus, the method reduces the amount of errors with additional semantic-scored pairs.

5 Conclusions

We have presented a ML-based system for RTE based on new similarity metrics as well as simple string-based metrics and a shallow-semantic metric. The new similarity measures are: i) Chunking, ii) Named Entities. The method has comparable performance with the average of methods in the RTE Challenges, but is far from the best and the related work. Our method relies in simple and few features, and our system just tackles one semantic phenomenon (i.e. 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 the performance increases. In order to improve the performance of our current ML system we can attempt to resolve the errors caused by the TINE metric based on the error analysis, or use a different semantic approach to RTE [10]. The semantic metric uses a distributional thesaurus to measure the similarity between arguments, and for example *cat* and *dog* will be aligned because they share the same context. One direction to improve the semantic metric is to add hard constraints over the core arguments, and these constrains can be defined as thresholds learned over the training dataset.

Acknowlegments

This work was supported by the Mexican National Council for Science and Technology (CONACYT), scholarship reference 309261.

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 rte3. 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. In: Proceedings of the 10th Mexican international conference on Advances in Artificial Intelligence - Volume Part I. pp. 267–276. MICAI'11, Springer-Verlag, Berlin, Heidelberg (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)