Methods for Measuring Semantic Similarity of Texts

Miguel Angel Rios Gaona

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton for the degree of Doctor of Philosophy

November 21, 2014

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of as- sessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

Signature: . . . . . .

Date: . . . . . .

Abstract

Measuring semantic similarity is a task needed in many Natural Language Processing (NLP) applications. For example, in Machine Translation evaluation, semantic similarity is used to assess the quality of the machine translation output by measuring the degree of equivalence between a reference translation and the machine translation output. The problem of semantic similarity (Corley and Mihalcea, 2005) is defined as measuring and recognising semantic relations between two texts. Semantic similarity covers different types of semantic relations, mainly bidirectional and directional. This thesis proposes new methods to address the limitations of existing work on both types of semantic relations.

Recognising Textual Entailment (RTE) is a directional relation where a text T entails the hypothesis H (entailment pair) if the meaning of H can be inferred from the meaning of T (Dagan and Glickman, 2005; Dagan et al., 2013). Most of the RTE methods rely on machine learning algorithms. de Marneffe et al. (2006) propose a multi-stage architecture where a first stage determines an alignment between the T-H pairs to be followed by an entailment decision stage. A limitation of such approaches is that instead of recognising a non-entailment, an alignment that fits an optimisation criterion will be returned, but the alignment by itself is a poor predictor for

non-entailment. We propose an RTE method following a multi-stage architecture, where both stages are based on semantic representations. Furthermore, instead of using simple similarity metrics to predict the entailment decision, we use a Markov Logic Network (MLN). The MLN is based on rich relational features extracted from the output of the predicate-argument alignment structures between T-H pairs. This MLN learns to reward pairs with similar predicates and similar arguments, and penalise pairs otherwise. The proposed methods show promising results. A source of errors was found to be the alignment step, which has low coverage. However, we show that when an alignment is found, the relational features improve the final entailment decision.

The task of Semantic Textual Similarity (STS) (Agirre et al., 2012) is defined as measuring the degree of bidirectional semantic equivalence between a pair of texts. The STS evaluation campaigns use datasets that consist of pairs of texts from NLP tasks such as Paraphrasing and Machine Translation evaluation. Methods for STS are commonly based on computing similarity metrics between the pair of sentences, where the similarity scores are used as features to train regression algorithms. Existing methods for STS achieve high performances over certain tasks, but poor results over others, particularly on unknown (surprise) tasks. Our solution to alleviate this unbalanced performances is to model STS in the context of Multi-task Learning using Gaussian Processes (MTL-GP) (Álvarez et al., 2012) and state-of-the-art

STS features (Šarić et al., 2012). We show that the MTL-GP outperforms previous work on the same datasets.

Acknowledgements

First, I would like to thank my supervisory team, Lucia Specia, Alexander Gelbukh and Ruslan Mitkov. Lucia who gave me the academic freedom to follow my ideas, her unconditional support during exhausting deadlines and her kindness to teach me how to conduct correct and sound research. Alexander who introduced me to the world of Natural Language Processing and his insightful lessons regarding research during my first years on postgraduate school. Ruslan Mitkov who gave me the opportunity and the support to carry out my studies in the university of Wolverhampton.

Next, I thank my friends who gave me the strength to accomplish this work. Wilker Aziz who was with me during the countless hours of working, gaming, having fun, etc. I am very lucky to have very special friends who make every day a great day: Iustin Dornescu, Marialaura Giova and Victoria Yaneva. I hope these great days will continue for a long time. I also thank the long-distance support from my friends in Mexico: Victor, Esmeralda and Luis Adrian.

Finally, I thank my parents for their unconditional support and love all these years that I was far away from home and as my mother always said "BE HAPPY!".

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

Acronyms

BoW bag-of-words

GP Gaussian Processes

ML Machine Learning

MLN Markov Logic Network

MT Machine Translation

MTL Multi-task Learning

NLP Natural Language Processing

NER Named Entity Recognition

PP Paraphrase Recognition

POS Part-of-Speech

QA Question Answering

RTE Recognising Textual Entailment

SUM Summarisation

STS Semantic Textual Similarity

SRL Semantic Role Labelling

SVM Support Vector Machine

TL Transductive Learning

IE Information Extraction

IR Information Retrieval

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the most important challenges in Natural Language Processing (NLP) is language variability: texts with the same meaning can be realised in several ways. NLP applications need to identify how their inputs and requested outputs are related, even if they have different surface forms, as they can express the same meaning. A way to address the language variability that can be explored across applications is the notion of semantic similarity. For example, semantic similarity serves as a criterion within Text Summarisation to select a sentence that summarises an entire paragraph (Das and Martins, 2007). In Machine Translation (MT) evaluation, semantic similarity estimates the quality of machine translations by measuring the degree of equivalence between a reference translation and the machine translation output (Banerjee and Lavie, 2005). The problem of semantic similarity is defined as measuring and recognising the presence of semantic relations between two texts (Corley and Mihalcea, 2005; Rus et al., 2013).

Semantic similarity is associated with different types of semantic relations, mainly directional and bidirectional (Rus et al., 2013). These two types of semantic relations provide different frameworks to address the se-

mantic needs of various NLP applications. It is common practice for work on directional relations to assign binary decisions to pairs of texts, while work on bidirectional relations assigns continuous decisions scores to pairs of texts. This thesis covers these widely studied instances of semantic similarity relations, namely Recognising Textual Entailment and Semantic Textual Similarity. Textual Entailment is a directional relation where a text T entails the hypothesis H (entailment pair) if the meaning of H can be inferred from the meaning of T (Dagan and Glickman, 2005; Dagan et al., 2013). On the other hand, Semantic Textual Similarity is defined as measuring the degree of bidirectional semantic equivalence between pairs of sentences (Agirre et al., 2012).

Recognising Textual Entailment (RTE) has been proposed as a generic task that captures major semantic inference (i.e. Textual Entailment) needs across many NLP applications (Dagan and Glickman, 2005; Dagan et al., 2013). In order to address the task of RTE, different methods have been proposed and most of these methods rely on supervised Machine Learning (ML) algorithms (Dagan et al., 2013). These approaches make the assumption that there is a relationship between high similarity scores and a *positive* entailment relation. Different sources of semantic information have been used for scoring similarity in RTE, such as lexical (e.g. Mehdad and Magnini (2009)), structural (e.g. Burchardt et al. (2007)) and coreference resolution (e.g. Mitkov et al. (2012)).

Another approach for RTE is to determine some sort of alignment between the T-H pairs. The hypothesis H is aligned with a portion corresponding to the text T, and the best alignment is used as a feature to train a classifier. A common limitation of alignment approaches is that instead of recognising non-entailment relations, they tend to choose an alignment that fits an optimisation criterion (de Marneffe et al., 2006). However, the use of alignment solely is a poor predictor of non-entailment relations.

In order to address this limitation, de Marneffe et al. (2006) propose to divide the process of textual entailment recognition into a multi-stage architecture, where the alignment and the entailment decision are separate stages. The alignment phase is based on matching graph representations of the T-H pair using dependency parse trees. For the entailment decision, de Marneffe et al. (2006) define rules that strongly suggest entailment relations. The specific rules between the T-H pair can be positive or negative, depending on whether they represent entailment or non-entailment.

In this thesis we also address the RTE problem by employing a multi-stage architecture. However, in contrast to previous work (de Marneffe et al., 2006), we have based both the alignment and entailment decision on semantic clues such as "Who did what to whom, when, where, why and how". These clues are given by a shallow semantic parser, namely a Semantic Role Labelling (SRL) tool. In particular, for the alignment stage we propose three new alignment methods to match predicate-argument structures between the T-H pairs. For the entailment decision stage we propose a propositional ML setup

based on using the output of the alignment method as features for building a classification model. We also propose a more advanced statistical model based on relational information extracted from the alignment stage. Instead of using simple similarity metrics to predict the entailment decision, the statistical relational learning model relies on rich relational features extracted from the output of the predicate-argument alignment structures between T-H pairs. A Markov Logic Network (MLN) learns to reward pairs with similar predicates and similar arguments, and penalise pairs otherwise.

In addition to RTE challenge datasets, we test our alignment methods on different applications such as MT evaluation and Semantic Textual Similarity (STS). On MT evaluation, we show that the addition of our alignment method to a common evaluation metric (i.e. BLEU) improves overall performance. However, for STS our predicate-argument alignment method shows poor results compared to simpler similarity metrics, and thus alternative methods were proposed.

STS measures the degree of semantic equivalence between two texts. STS as an instance of a bidirectional semantic equivalence is related to RTE, but it is more directly applicable to NLP applications such as Question Answering (Lin and Pantel, 2001b), Text Summarisation (Lin and Hovy, 2003) and Information Retrieval (Park et al., 2005), which depend directly on measuring the degree of semantic similarity between pairs of texts. STS differs from RTE in that it assumes a graded equivalence between the pair of texts. For

example, in RTE a *car* is a *vehicle*, but a *vehicle* is not a *car*. On the contrary, in STS a *vehicle* and a *car* are more similar than a *computer* and a *car*.

Methods for STS are commonly based on computing similarity metrics between pairs of sentences, where the similarity scores are used as features to train regression algorithms. For example, Šarić et al. (2012) extract features from similarity metrics based on word overlap and syntax similarity. As in most methods, a separate model is built for each one of the NLP tasks (i.e. applications), such as Machine Translation evaluation and video paraphrasing. As result, STS methods show unbalanced performances across tasks. These methods also present poor generalisation on new unseen test tasks. In order to address this limitation, Heilman and Madnani (2013) propose to incorporate domain/task adaptation techniques (Daumé et al., 2010) for STS to generalise models to new tasks. In the context of STS previous work focus on leverage information among tasks (i.e. task adaptation). They add new features into the model, where the feature set contains task specific features plus general task features. When an instance of a specific task is to be predicted, only the copy of the features of that task will be active; if the task is unknown, the general features will be active. Severyn et al. (2013) propose to use meta-classification to cope with task adaptation. They merge each text pair and extract meta-features from them such as bag-of-words and syntactic similarity scores. The meta-classification model predicts, for each instance, its most likely task based on the previous features. The contribution of these task adaptation techniques to STS is not clear, given that they do not im-

prove overall performance and require that a specific model is assigned to an unseen test task (Heilman and Madnani, 2013).

We propose a more advanced task adaptation technique using Gaussian Processes (GP) in an Multi-task Learning (MTL) setting to achieve balanced performance and generalised learning across task. We use a state-of-the-art STS feature set (Šarić et al., 2012) and show that the MTL model improves the results of other methods using the same feature set on the same datasets. In addition, we use an MTL-GP model based on a combination of kernels to tackle task adaptation. The combination of kernels learns general and task-specific information for unknown and known test tasks, respectively. Our method achieves superior or at least comparable performance compared to previous work based on task adaptation.

1.1 Contributions

The main contributions of this thesis are new methods for both types of semantic relations: RTE and STS.

In previous work (de Marneffe et al., 2006), RTE is divided into two stages, where the first stage is based on the use of an alignment technique and the second stage uses heuristics to determine the entailment decision. However, these heuristics are pre-defined, handcrafted rules or features based on the intuition of what an entailment should be realised. This leads us to our first research question:

To what extent the relational information extracted from semantically aligned T-H pairs affects the performance of an RTE method?

In order to answer the first research question we explore different alignment methods, and different entailment decision schemes based on the model of a multi-stage architecture for RTE. We aim to encode the entailment decision intuition directly into the feature design of the second stage with a statistical relational learning model. The intuition behind our RTE method is that an aligned T-H pair with similar situations and similar participants is likely to hold an entailment relation. Our contributions belong to the alignment and the entailment decision stages. Statistical relational learning (Getoor and Taskar, 2007), as opposed to a propositional formalism, is focused on representing and reasoning over domains with a relational and a probabilistic structure. These models use first-order representations to describe the relations between the domain variables and probabilistic graphical models to reason over uncertainty (Richardson and Domingos, 2006). The MLN framework encodes relational information about the domain under study. For example, instead of creating fixed rules, we can encode our intuition about the entailment decision stage into the feature design with soft constraints that penalise or reward T-H pairs according to relations between predicates and arguments.

In this work, we provide experimental support for the above research question with the following contributions:

1. New alignment methods based on the matching of predicate-argument structures (Chapter 3). The first method uses ontologies and distributional information for partial matching of predicates and arguments. The findings were published in (Rios et al., 2011).

   The second method is based on an optimisation step to match predicates which accounts for context (i.e. similar arguments) and how the predicates are related. The findings were published in (Rios et al., 2012) and (Rios and Gelbukh, 2012a). Both methods show average performance on RTE datasets.

   The last method is a modification of the edit distance method (Kouylekov and Magnini, 2005) to leverage predicate argument information within the distance metric. Our edit distance method also led to average performance on RTE datasets. The findings were published in (Rios and Gelbukh, 2012b).

2. We propose a statistical relational learning model for the decision stage (Chapter 3). The method achieves the best results for the RTE-3 dataset and shows comparable performance against the state-of-the-art methods for other datasets. The findings were published in (Rios et al., 2014).

3. We reformulate a first version of the alignment method as a similarity metric to fit the task of MT evaluation (Chapter 4). Our method shows performance that is comparable to that of other metrics on MT evaluation at segment level for several language pairs. We show that the addition of the alignment method improves the performance of metrics such as BLEU. The findings were published in (Rios et al., 2011).

4. We use a second version of the alignment method for the task of STS (Chapter 4). We use the method as a similarity metric to train a regression algorithm. Compared to the official results of the first STS evaluation challenge, our method ranks above average, but the contribution of the semantic metrics to the STS task is poor. The findings were published in (Rios et al., 2012)

Previous work on STS achieve good results on certain tasks, but poor results on others (e.g. Šarić et al. (2012)). Moreover, these methods have to cope with the challenge of missing training data for unknown (surprise) tasks. Given the results of our alignment methods and the challenges observed in previous work on STS, we formulate our second research question:

To what extent the simultaneous learning of multiple related tasks affects the overall performance of an STS method?

We analyse the second research question with the hypothesis that an STS model based on MTL can improve the overall performance by learning models

for different, but related tasks simultaneously, compared to learning models for each task separately. MTL is based on the assumption that related tasks can be clustered and inter-task correlations can be transferred. Our contribution belongs to reducing the gap in terms of results across tasks observed in previous work. We do that by using state-of-the-art features to cope with the limitations of our alignment method and an MTL algorithm. We use as model a non-parametric Bayesian approach based on kernels, namely GP (Rasmussen and Williams, 2005).

We provide experimental support for the second research question with the following directions:

1. We use MTL to cope with unbalanced performances and unknown tasks (Chapter 4). The MTL model outperforms previous work on the 2012 STS evaluation challenge, achieves robust performance on the 2013 evaluation challenge datasets (i.e. unknown test tasks) and competitive results on the 2014 dataset. The findings were published in (Rios and Specia, 2014).

2. We use use a linear combination of kernels to generalise learning on unseen test tasks. This model gives us the control of the inter-intra task transfer by choosing which kernel to use for each known or unknown test task. The linear combination model outperforms previous work based on task adaptation for most of the 2013 datasets.

1.2 Organisation of the Thesis

The organisation of this work is as follows:

- In Chapter 2, we present a literature review on RTE and STS. We describe the main models for RTE: i) Logic-based Inferencing, ii) Similarity Metrics, iii) Transformation Sequences. We then describe the main components in STS: i) Similarity Metrics and ii) Task Adaptation.

- In Chapter 3, we describe the proposed RTE methodology, including the new alignment methods, and the statistical relational learning model for RTE. We provide the experimental settings used to apply the new methods for RTE and compare our results against related work.

- In Chapter 4, we test the proposed alignment methods on MT Evaluation and STS. Furthermore, we describe the STS method based on MTL. We describe the experimental settings used to apply the new method for STS. Finally, we compare our results with related work.

- In Chapter 5, we discuss the conclusions, revisit our contributions and describe further research directions.

Chapter 2

Background

In this Chapter, we present a survey of the models for both types of semantic similarity relations. We begin by describing the models for the textual entailment (i.e. directional relation). The RTE task consists of deciding, given two text expressions, whether the meaning of one text T is entailed from the meaning of the hypothesis H. Dagan and Glickman (2005) give the definition for textual entailment as:

We say that the text T entails the hypothesis H (entailment pair) if the meaning of H can be inferred from the meaning of T as could typically be interpreted by people.

Based on the definition of applied textual entailment, in 2005 the PASCAL Network of Excellence[1] started the RTE Challenge (Dagan and Glickman, 2005), which provides the benchmark for the RTE task. The participant methods decide for each entailment pair whether T entails H or not. The annotation used for the entailment decision is *TRUE* if T entails H or *FALSE* otherwise.

RTE can be thought as a classification problem, where the entailment relations are the classes, and the RTE datasets provide the essential evidence

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE/

to build a supervised binary classifier (Dagan et al., 2010). The variations of ML-based methods for RTE depend on the model used in order to train the supervised binary classifier. In other words, the representation (e.g. words, syntax, semantics) of the T-H pair that is used to extract features to train a supervised classifier. For example, a baseline method proposed by Mehdad and Magnini (2009) consists of measuring the word overlap between the T-H pairs, where the word overlap is the number of words shared between the text and the hypothesis. The method is divided into three main steps: i) pre-processing: All T–H pairs are tokenised and lemmatised, ii) computing the word overlap, and iii) building a binary classifier. An overlap threshold is automatically learnt from the training data, and then the test data is classified based on the learnt threshold. If the word overlap (i.e. similarity) score is greater than the threshold the entailment decision is TRUE, otherwise it is FALSE. Thus, this method is based on the assumption that a relation exists between high similarity scores with a positive entailment (i.e. TRUE), and also between low similarity scores with a negative entailment (i.e. FALSE).

There are three classes of models in RTE:

Logic-based Inferencing The T-H pair is represented with symbolic meaning representations and a theorem prover tries to search for a proof. The proof is added as a binary feature (i.e. presence or absence of the proof) into the binary classifier. The motivation for this model is that a theorem prover can find a formal proof by using background knowl-

edge and a first-order logic formula (e.g. symbolic representation of the text).

Similarity Metrics The T-H pair is represented by similarity scores computed from different linguistic levels. These scores become features used to train the binary classifier. The motivation for this model is that a pair with a strong similarity holds a positive entailment relation.

Transformation Sequences The T-H pair is represented by a linguistic level annotation (e.g. syntax trees), and a series of transformations are applied to transform T into H. It is hypothesised that the smaller the amount of transformations is, the stronger the positive entailment relation. The amount of transformations becomes a distance score which is used as a feature to train the classifier.

In the remainder of this section we show the most relevant work for each one of the above types of methods. For a comprehensive description of RTE models we refer the reader to (Dagan et al., 2013).

2.1 Recognising Textual Entailment Methods and Techniques

2.1.1 Logic-based Inferencing

The motivation behind this model is that the method performs a search for whether or not the entailment holds by finding proofs with a theorem prover.

Despite the strong theoretical foundation of these methods they do not work well in practice. This is mostly due to the lack of background knowledge which not many true decisions could be found.

Bos and Markert (2005) propose an ML-based method by combining shallow and deep features. The shallow feature comes from a simple lexical overlap between the words of the T-H pair, and the deep feature comes from the output of a theorem prover. In order to provide the input for the theorem prover first the T–H pair is transformed onto a knowledge representation based on Discourse Representation Theory (Kamp and Reyle, 1993), and this representation is mapped into first-order logic. A theorem prover and a model builder are used as inference engines. The theorem prover tries to find a proof for the input. The axioms used to support the proof for the theorem prover are extracted from WordNet (Fellbaum, 1998) and the CIA fact book (geographical knowledge). The model builder tries to find a model with the negation of the input. If the models do not differ too much in size (a model is the number of propositions generated by the inference engine), it is likely that the entailment relation holds, since H does not introduce any, or little, information into the model. To combine the shallow and deep features, a decision tree is trained on the development dataset of the RTE. The method is able to parse semantic representations and then search for proofs for 774 of all 800 T-H pairs in the test data (a coverage of 96.8%). The theorem prover only finds 30 proofs of which 23 are annotated as entailment in the gold standard. Bos and Markert (2006) improved their previous method

by using the following model builders: Paradox[2], and Mace[3], and new deep semantic features such as first-order logic formulas for entailment and the entailment formulas with the addition of background knowledge formulas, which are used with the theorem prover.

In contrast, Fowler et al. (2005) develop COGEX, which is a modified version of the OTTER [4] system (theorem prover). This modified version is adapted to work with natural language expressions. The method uses as input a list of clauses (set of support) used for the inference search. The set of support clauses is loaded into the method along with the negated form of H (proof by refutation) as well as T. A second list that contains clauses (handcrafted axioms) is used by the method to generate the inferences. This list consists of axioms generated by hand or automatically. The axioms are used to provide background knowledge as well as syntactic knowledge extracted from WordNet lexical chains. First, the method parses each T–H pair into a first-order logic (Moldovan and Rus, 2001) representation. Second, the method generates formulas from the first-order logic representation to be solved by a theorem prover. The background knowledge used to support the proof consists of 310 common-sense rules, linguistic rewriting rules and WordNet lexical chains. The lexical chain is a chain of relations between two WordNet synsets (e.g. hyponym, hypernym, synonym relations). For each relation in the chain the method generates an axiom. For example, the

[2]http://vlsicad.eecs.umich.edu/BK/Slots/cache/www.cs.chalmers.se/ koen/paradox/
[3]http://www.mcs.anl.gov/research/projects/AR/mace/
[4]http://www.mcs.anl.gov/research/projects/AR/otter/

chain "buy" → "pay". The axiom states that the predicate from the first synset implies the predicate in the second synset. The COGEX theorem prover searches for proofs by weighting the clauses (the negated H has the largest weight in order to ensure that it will be the last to participate in the search). COGEX removes the clause with the smallest weight from the set of support clauses, and it searches in the second list for new inferences. All produced inferences are assigned an appropriate weight depending on what axiom they were derived from and appended to the set of support list. COGEX repeats the previous steps until the set of support is empty. If a refutation is found, the proof is complete. If a refutation cannot be found, the weights are relaxed. When a proof by refutation is found, a score for that proof is calculated by starting with an initial perfect score and deducting points for axioms that are utilised in the proof, weights that are relaxed and predicates that are dropped. The score computed by COGEX is only a metric of the axioms used in the proof and the significance of the dropped arguments and predicates. The confidence score (entailment decision) for a T-H pair is measured as the distance between the score and the threshold. The threshold is learnt from the benchmark dataset.

Bayer et al. (2005) divide the entailment decision into two stages. First, the alignment of the T-H pair, and second, an inference stage. The alignment stage is based on GIZA++ [5], which uses the Gigaword corpus to extract word statistics. The alignment is based on the hypothesis that the

[5]http://code.google.com/p/giza-pp/

newspaper headlines are often aligned (i.e. entailed) by the corresponding lead paragraph. The inference stage takes the aligned pairs, and extract from them the following features: tokens, part-of-speech tags, morphology and syntactic relations (Link Grammar and Dependency Parsing). The logic feature is based on extracting information about the events from the text (Bayer et al., 2004). Finally, the logic representation is used as input in a probabilistic inference engine (Epilog[6]). Epilog is an event-oriented probabilistic inference engine. The input for Epilog consists of the above features, and the entailment decision is based on a Support Vector Machine (SVM) classifier. However, the method fails to prove entailment for almost all the T-H pairs. Because of parser mistakes, the method fails to convert 213 out of the 800 test pairs into the event logic.

MacCartney and Manning (2007) propose a framework called Natural Logic for RTE. Natural Logic is similar to a first-order logic representation of the T-H pairs, where the method transforms T into H based on a low cost edition scheme. Thus, it learns to classify entailment relations based on the cost of atomic edits. The method first extracts the syntactic trees of the T-H pairs. Second, it computes the monotonicity between the constituents of the syntactic trees, where the monotonicity is based on the semantic types theory from Montague in (Thomason, 1974). A relation is monotonic if one node is a generalisation of another node, where a hypernym relation is an example of generalisation. Third, it computes the alignment (Cooper et al., 1996) be-

[6]http://www.cs.rochester.edu/research/epilog/

tween the T-H pair annotated with the previous monotonic constituents. The method makes the entailment classification based on an ML algorithm. The classification is based on alignment, and the method allows finding deeper semantic alignments. With the addition of Natural Logic the method improves the overall performance. The semantic alignment is computed over dependency parse trees, where the patterns to be aligned come from regular expressions. For example, {word:run;tag:/NN/} refers to any node in the tree that has a value *run* for the attribute *word* and a *tag* that starts with *NN*, while {.} refers to any node in the dependency tree. The Natural Logic method does not translate natural language into a first-order logic, where the proofs are expressed as edits to natural languages expressions. The edits are done at conceptual level, such as contractions and expansions. For example, the model defines an entailment relation (i.e. positive entailment $\rightarrow$) between nouns (hammer $\rightarrow$ tool), adjectives (deafening $\rightarrow$ loud), verbs (sprint $\rightarrow$ run), modifiers, connectives and quantifiers. In upward-monotone contexts (i.e. the H tends to be a generalisation), the entailment relation between compound expressions uses the entailment relations between their parts. Thus, "tango in Paris" $\rightarrow$ "dance in France", since "tango" $\rightarrow$ "dance" and "in Paris" $\rightarrow$ "in France".

In summary, although the Logic-based Inferencing approaches are based on a strong theory they use expensive processing tools and their results do not outperform simpler methods (e.g. similarity metrics as features for super-

vised ML algorithms). These types of methods have practically disappeared from the recent RTE literature.

2.1.2 Similarity Metrics

The motivation behind this model is that a pair with a strong similarity score holds a positive entailment relation. Different types of similarity metrics are applied over the T-H pair to extract features and then train a classifier. As we mentioned before a baseline method consists of: a linguistic preprocessing of the T-H pair, computing similarity metrics between the T-H pair, training a supervised binary classifier with the feature space and finally, classifying new T-H pairs.

The preprocessing extracts different linguistic representations (e.g. lemmas, syntactic trees, symbolic representation, etc.) of the text. Thus, each linguistic representation has different operations to measure similarity between their structures. For example, if the representation of the T-H pair is bag-of-words (BoW) an operation used to measure similarity could be string similarity metric such as BLEU (Papineni et al., 2002). The configuration used for BLEU is the following: the text T is the *reference* translation and the hypothesis H is the *candidate* translation, the entailment decision is based on an empirical set threshold, and if the BLEU score is above this threshold the entailment decision is TRUE, otherwise it is FALSE (Pérez and Alfonseca, 2005).

Moreover, BLEU is not the only metric used to measure similarity. Malakasiotis and Androutsopoulos (2007) train a SVM using different string similarity metrics as features. They propose to decide the entailment relation by using similarity metrics such as: Jaro-Winkler, Soundex, Manhattan distance, Euclidean distance, N-gram distance, matching coefficient and Jaccard coefficient. Finally, in order to outperform the results by just using string similarity metrics the authors apply a feature extraction techniques for each task (Information Retrieval (IR), Information Extraction (IE), Question Answering (QA), Summarisation (SUM)). They show that the use of task specialised ML models improve the overall performance. In other words, a model should be trained for each task individually.

However, a BoW approach just can tackle certain entailment phenomena. Vanderwende et al. (2005) shows the contribution of syntax for RTE. The experiment relies on human annotators to decide if the information from an idealised parser is enough to decide the entailment relation. They show that syntax can handle 34% of the test pairs, and with the addition of a thesaurus the coverage grows up to 48%.

Different representations and world knowledge are needed to increase the coverage of a method based on similarity metrics. Pazienza et al. (2005) propose to measure the distance between syntax trees by using a SVM. They define the entailment as a subsumption relation between the text and the hypothesis. First, the text semantically subsumes the hypothesis. For example, from the text "The cat eats the mouse" follows the hypothesis "The

cat devours the mouse", where *eat* is a semantic generalisation of *devour*. Second, the text syntactically subsumes the hypothesis. For example, from the text "The cat eats the mouse in the garden" follows the hypothesis "The cat eats the mouse". The text contains a prepositional phrase. Finally, the text implies directly the hypothesis. For example, the hypothesis "The cat killed the mouse" is implied by the text "The cat devours the mouse", as it is supposed that *killing* is a precondition for *devouring*. The T-H pairs are represented by subject-verb-object representation, and the threshold is learnt via a SVM. The subsumtion relations are based on measuring the edges of the graphs for the subject-verb-object relation, and on measuring the nodes for the semantic relations.

Although the SVM classifier is a common choice for binary classification, Inkpen et al. (2006) experiment with different supervised ML algorithms. The features used to train the classifier are: lexical, syntactical (i.e. dependency relations from Minipar), mismatching of negations and numbers. The ML algorithms used for the classification are: decision trees, Naive Bayes, SVM and K-nearest neighbours. They show that the best ML algorithm, given the previous features, is the SVM.

The BoW and syntax representations lack semantic information (i.e. background knowledge). For example, the hypernym relation "cat" $\rightarrow$ "animal" can not be decided by measuring word similarity. Thus, the string similarity metrics can not address lexical relations extracted from an ontology such as: synonymy, hypernymy, hyponymy, and so on. Jijkoun and de Rijke

(2005) propose the use of WordNet similarity metrics for RTE. The metrics are: Dekang Lins dependency-based word similarity (Lin, 1998a), and lexical chains in WordNet (Hirst and St-Onge, 1997).

In contrast, Burchardt and Frank (2006) propose the use of a deep semantic analysis based on graph matching and ML algorithms. This method relies on a different type of semantic formalisms. The formalisms used to represent the graphs are the Lexical Functional Grammar (Crouch and King, 2006) and Frame Semantics (Baker et al., 1998). The method differs from the method of Bos and Markert (2005), that is, a fine grained semantic analysis and reasoning method, by achieving a high recall but at the cost of a low precision. Burchardt and Frank (2006) defines the semantic analysis as a structural and semantic overlap over the Frame Semantics structures.

Burchardt et al. (2007) introduce a method, which involves deep linguistic analysis and shallow word overlap. The method consists of three steps: first, representing the T-H pair with the Frame Semantics and Lexical Functional Grammar formalisms (this representation is similar to SRL). Second, extracting a similarity score based on matching the Lexical Functional Grammar graphs and then making a statistical entailment decision. Burchardt et al. (2007) use previous RTE 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 SRL for RTE use the annotation provided by a semantic parser to measure the similarity between texts.

These methods only measure the similarity in terms of how many labels the structures share (overlaps) and not the content of those labels.

Different sources of semantic information such as lexical (e.g. WordNet) or structural (e.g. SRL) have been used for RTE. However, the role of discourse information in RTE is limited because of the format of the T-H pairs (i.e. short T-H pairs with 38 words in average). Before the third RTE-3 challenge dataset, the format of the T-H pair (i.e. size of a paragraph for the T text) was more suitable to extract discourse information. Castillo (2010) proposes the first attempt to use discourse information in the context of the RTE Search Task. The method transforms the documents into the standard T-H pair format by using coreference chains. The sentences related to an entity found in a coreference chain are incrementally appended to T, and this is computed for each entity in the document. A standard ML algorithm (i.e. SVM) based on string similarity metrics is applied to decide the entailment relation. Previous work on the impact of discourse information on RTE includes: Delmonte et al. (2007), who study the the coverage of anaphora phenomena on the RTE datasets, and Andreevskaia et al. (2005), who propose a method for paraphrase detection based on coreference resolution, where the use of discourse information hardly improves the overall performance. (Mitkov et al., 2012) study the impact of coreference resolution on NLP applications, where the contribution to RTE is not statistically significant.

Mirkin et al. (2010) argue that discourse information can improve the overall performance of RTE in the context of the Search Task. They manually analyse a part of the Search task dataset to measure the impact of coreference resolution on RTE. The study shows that most of the discourse references that decide the entailment relation are: nominal coreference, and verbal terms. The substitution of the referents with the entities itself is not enough to extract all the information from discourse references. Mirkin et al. (2010) suggest that the discourse information should be integrated into the inference engine and not simply be part of a pre-processing step or a shallow feature for a supervised ML algorithm.

Delmonte et al. (2005) propose an approach based on measuring the semantic similarity between the T-H pairs. The features are as follows: tokens, morphology, named entities, part-of-speech, syntax and SRL. The SRL feature is based on measuring the dissimilarities between the T-H pair. The shallow features (e.g. tokens and part-of-speech) basically score the overlap between the different representations of the T-H pair. The SRL similarity consists of checking the mismatch over:

- presence of spatio-temporal locations to the same governing predicate
- presence of opacity operators such as discourse markers for having conditionality a scope over the same predicate
- presence of quantifiers and other type of determiners attached to the same noun phrase head in the T-H pair under analysis

- presence of antonyms in the T-H pair at the level of predicates

- presence of predicates belonging to the class of "doubt" expressing verbs

Delmonte et al. (2005) 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. (2007) show improvements to the previous method by introducing additional modules, each of which uses fine grained inferential triggers such as anaphora resolution and the matching of grammatical relations. They show that the RTE-3 dataset contains pairs that could be answered via anaphora resolution, from a total of 800 pairs in the development and test datasets: 117 pairs in the test dataset and 135 pairs in the development dataset.

Andreevskaia et al. (2005) represent the T–H pair as shallow predicate-argument structures. The predicate-argument structures are extracted from the RASP parser[7]. The method recognises paraphrases based on coreference, and it measures distance between verbs using WordNet lexical chains. If the hypothesis H contains the pattern "X is Y" (part of a predicate-argument). If X is in H and X is in the text T, therefore the pair X, X' belongs to the same inter-sentence coreference chain. If Y is in H and Y is in T, therefore

[7]http://www.informatics.susx.ac.uk/research/groups/nlp/rasp/

the pair Y, Y' belongs to the same inter-sentence coreference chain. If X corefers with Y the pair is a paraphrase.

de Marneffe et al. (2006) also use a two-stage alignment, but they use dependency trees as representation instead of SRLs for the alignment. Chambers et al. (2007) improve the alignment stage used in (de Marneffe et al., 2006) and they combine it with a logical framework for the second stage (MacCartney and Manning, 2007). MacCartney et al. (2008) propose a phrase-based alignment that uses external lexical resources. Glickman and Dagan (2006) 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.

Garrette et al. (2011) combine first-order logic and Statistical Relational Learning methods for RTE. The approach uses discourse structures to represent T-H pairs, and an MLN model to perform inference in a probabilistic manner over the following semantic phenomena: implicativity, factivity, word meaning and coreference. A threshold to decide the entailment 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. Beltagy et al. (2013) extend the work in (Garrette et al., 2011) to be able to process large scale datasets such as those from the RTE challenges.

In summary, the methods which use semantic features address a wide variety of phenomena, but simple methods that use lexical features are a difficult baseline to defeat. For example, Litkowski (2006) uses a simple overlap

metric for RTE that shows very strong results compare to state-of-the-art methods. Litkowski (2006) argues that syntactic and semantic tests do not appear to improve overall results, and WordNet does not provide adequate levels of granularity. Litkowski (2006) proposes that possible improvements can be obtained from a thesaurus and syntactic alternation patterns derived from FrameNet.

2.1.3 Transformation Sequences

The motivation behind this model is that the entailment relations can be measured by applying series of transformations of T into H. This means that if the cost of a series of transformations over T is low, T is similar to H and they hold a positive entailment relation.

Edit distance algorithms are a common approach to transform texts, where the basic edit operations are: insertion, substitution and deletion. Each operation has an attached score, which means that some operations are more expensive than others, and this cost is usually learnt via supervised ML algorithms. The edit distance algorithms score the difference between a pair of texts based on how many operations were necessary to transform T into H. Kouylekov and Magnini (2005) introduce the edit distance algorithms for RTE. The assumption is based on estimating the cost of the information of the hypothesis, which is missing in the text. The T-H pair holds an entailment relation if there is a sequence of operations over T that produce H with an overall cost below a certain threshold. The threshold, as well as the

cost of each operation, are learnt from the development dataset by using ML techniques.

Cabrio et al. (2008) describe a framework that consists of a combination of specialised entailment engines each one addressing a specific entailment phenomenon. Due to the fact that RTE is a combination of several phenomena, which interact in a complex way. Each engine is trained to deal with a different aspect of language variability (e.g. syntax, negation, modal verbs). Also, this framework has a modular approach to evaluate the progress on a single aspect of entailment using the training data. The entailment engines are based on edit distance algorithms. In each engine the cost of each edit operation is defined (learnt) according to a specific phenomenon. The cost schemes of the different engines are defined in order not to intersect each other. If the costs of the edit operations are set as not 0 for a certain phenomenon, they are set as 0 for the aspects that are considered by another engine.

Transformation approaches can be combined with ML standard techniques. Roth and Sammons (2007) use semantic logical inferences for RTE, where the representation method is a bag-of-lexical-items. The bag-of-lexical-items 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 bag-of-lexical-items (auxiliary verbs, articles, exclamations, discourse markers and words in WordNet). Moreover, in order to recognise relations over the T-H pairs the method

checks matches between SRLs, and then applies a series of transformations over the semantic representations making it easier to determine the entailment. The transformation operations are: *annotate* makes some implicit property of the meaning of the sentence explicit. *Simplify/Transform* removes or alter some section of T in order to improve annotation accuracy or to make it more similar to H. *Compares* (some elements of) the two members of the entailment pair and it assigns a score that correlates to how successfully (those elements of) the H can be subsumed by T.

Harmeling (2007) propose a model that computes entailment decisions with a certain probability given a sequence of transformations over a parse tree. Wang and Manning (2010) merge the alignment and the decision into one step, where the alignment is a set of latent variables. The alignment is used into a probabilistic model that learns tree-edit operations on dependency parse trees.

The entailment relation can be also defined by matching entailment rules (i.e. positive or negative entailment) against the T-H pair. A widely used knowledge base for entailment rules is DIRT (Discovery of Inference Rules from Text) (Lin and Pantel, 2001a), which is based on the distributional hypothesis (Harris, 1954). The distributional hypothesis states that words with similar contexts tend to have the same meaning. Lin and Pantel (2001a) add the following extension: paths in dependency trees that occur in similar contexts tend to have similar meanings. For example, the template rule "Y is solved by X" $\rightarrow$ "X resolves Y", where, X and Y can have any surface

realisation, and the knowledge base has relations of the type: entailment and paraphrasing.

Nielsen et al. (2006) use lexical features based on similarity scores such as unigrams, bigrams and stems. The method uses the DIRT knowledge base as a set of syntactic templates (i.e. entailment rules). Thus, the entailment decision is not just given by the similarity between the T-H pair, it is also given by matching the DIRT templates. This means that the matching of a rule becomes an evidence to decide a binary feature. Nielsen et al. (2006) train different classifiers for each task: IE, IR, QA and SUM.

Zanzotto et al. (2006) propose to learn entailment relations from positive and negative entailment examples. The approach is similar to DIRT, but the extracted rules are focused just on entailment. First, the method parses each T-H pair. Second, the method sets anchors in the trees. The anchors are content words that maximise a given WordNet similarity metric. With these anchors the method searches for common subtrees between the T and H pairs. The common subtrees form a set of templates both positive and negative. For example, the rule: "X (VP (V ...) (NP (to Y)...)" $\nrightarrow$ "X is Y", is applied to the T-H pair: "Jacqueline B. Wender is Assistant to the President of Stanford University." $\nrightarrow$ "Jacqueline B. Wender is the President of Stanford University.". Zanzotto et al. (2007) propose a shallow-semantics fast rule learner that acquires rewriting rules from examples based on cross-pair similarity. The rules are placeholders between sentence words that are co-indexes in two substructures in parse trees of the T-H pair.

Marsi et al. (2007) use the DIRT dataset for RTE. The method is based on paraphrase substitution. If a T-H pair subtree is a paraphrase which has the same syntactic path of the DIRT dataset the entailment relation is TRUE otherwise it is FALSE.

Bar-Haim et al. (2007) propose a knowledge-based inference method for RTE. The method proofs (transforms pairs of text in the same fashion as an edit distance method) syntactic trees. In other words, the method transforms the T syntactic tree into the H syntactic tree. The analogy to logic-based proof methods is that generating a target text from a source text using the entailment rules will be the same process as a theorem prover. The T-H pair is represented as T (the left hand side) and H (the right hand side). If a part of the left and right hand sides is matched with an entailment rule from a knowledge base, the entailment relation is supported by the relation of the rule. For example, the pair "I visited New York" → "I visited a city" matches with the rule "New York" → "city". The entailment rules can be lexical, syntactic or semantic (e.g. SRL), and the rules are extracted from large corpora based on the distributional hypothesis (Harris, 1954). The motivation for this method is that the entailment relation can be defined (or supported) by matching rules against the T-H pair in addition to the transformation approach.

Magnini et al. (2014) propose an open source software containing state-of-the-art algorithms for RTE, the EXCITEMENT Open Platform (EOP). The EOP implements a generic architecture for multilingual textual entail-

ment (i.e. English, German and Italian). The platform implements several annotation pipelines, similarity metrics, knowledge resources and different entailment decision algorithms (Bar-Haim et al., 2007; Cabrio et al., 2008; Wang and Neumann, 2007).

In sum, the Transformation Sequences methods are an alternative for expensive theorem provers, and most of them rely on syntactic representations. These methods outperform logic-based methods in terms of accuracy and recall.

2.2 Semantic Textual Similarity Methods and Techniques

STS measures the degree of semantic equivalence between two sentences (Agirre et al., 2012). The STS evaluation campaign uses datasets that consist of pairs of texts from NLP tasks such as paraphrasing, video paraphrasing and machine translation evaluation. The participating methods have to predict a graded similarity score from 0 to 5. For example, a score of 0 means that the two texts are on different topics and a score of 5 means that the two texts have the same meaning.

We can broadly classify the models for STS given previous work as follows:

Similarity Metrics The sentence pairs are represented as vectors of similarity features used to train regression algorithms.

Task Adaptation The methods use domain/task adaptation techniques to cope with the challenge of unknown tasks.

In what follows, we describe the top performing methods as well as methods related to our work. For a complete description of the methods for STS we refer the reader to (Agirre et al., 2012, 2013, 2014).

2.2.1 Similarity Metrics

The motivation behind this model is that a similarity score between a pair of texts is correlated with the human annotations in terms of the degree of semantic equivalence.

Bär et al. (2012) use similarity metrics of varying complexity. The range of features goes from simple string similarity metrics to complex vector space models. The method shows the highest scores in the official evaluation metrics. The method does not achieve the best results in individual tasks but it is the most robust on average. It is worth mentioning that the method uses a state-of-the-art textual entailment approach (Stern and Dagan, 2011) for generating entailment scores to serve as features. However, the contribution of the textual entailment features is not conclusive given that they were not chosen by a feature selection algorithm.

Šarić et al. (2012) use a similar set up based on extracting features from similarity metrics such as: word overlap, WordNet path metric, alignment metric, vector space metric and syntactic similarity. Their method was

among the best performing ones in the paraphrasing datasets. In machine translation datasets, however, their method did not show a satisfactory performance beyond the training phase. Šarić et al. (2012) claim that differences between the train and test MT datasets in terms of length and word choice show that the MT training data is not representative of the test set for their choice of features, where for each dataset the method uses a separate Support Vector Regression (SVR) model. The results show that the word overlap, WordNet path metric and the alignment metric obtain high correlation despite the individual SVR models. The other features were shown to contribute to the individual SVR models.

Jimenez et al. (2012) propose to modify a cardinality function (e.g. Dice coefficient) for STS. The modified function is based on soft cardinality instead of set cardinality. Cardinality methods in general count the number of elements that are not identical in a set, while soft cardinality uses an auxiliary inter-element similarity function to make a soft count. The intuition for computing the soft cardinality is to treat elements in a set as sets themselves and to treat inter-element similarities as the intersections between the elements.

Heilman and Madnani (2012) propose a method based on discriminative learning, where the main contribution resides on modifications to the TERp (Wang et al., 2005) metric for machine translation evaluation. The modifications allow to use the metric beyond the scope of MT evaluation. The method shows competitive performance for the unknown tasks. For a pair of

sentences TERp finds the set of edit operations that convert one sentence into the other and then it produces a score. However, the features used in TERp make the metric difficult to apply to other tasks. For example, the one-to-one alignment of edit operations and the use of a greedy learning algorithm. The modified TERp uses the inference algorithm from the original metric to find the minimum cost sequence of editions, in contrast to the original inference algorithm, the modified method uses a discriminative algorithm to learn the weights for the edit-cost. The modifications are used as additional features. The corresponding parameters are learnt by the discriminative algorithm as opposed to the heuristic used by the original metric.

Following the intuition of using simple and MT metrics for similarity prediction, Yeh and Agirre (2012) propose a method focused on simple metrics. The semantic similarity metrics are based on ontologies, lexical matching heuristics and part-of-speech tagging metrics. The MT complementary metric is BLEU. Moreover, the method deals with unknown datasets by combining all the training data to train an unified model. The error analysis shows a high performance variation across the test datasets.

Banea et al. (2012) propose a synergistic approach to STS. The method uses semantic similarity metrics to train a supervised regression model. For the known test tasks there are individual models trained for the corresponding training task, but for unknown task the training dataset consists of all the training instances. The method shows a robust performance by combining all training data from different task into a single model. Moreover,

the corpus-based metrics (i.e. Distributional semantics) show a higher contribution to the overall performance than the knowledge-base metrics (i.e. WordNet path metrics).

Han et al. (2013) propose a new semantic similarity feature based on a combination of two metrics: distributional similarity and WordNet path similarity. The method also uses a simple alignment algorithm, which penalises poorly aligned words. The new semantic similarity feature rewards the distributional similarity score if the method finds relations between words such as: i) words that are in the same WordNet synset, ii) words have a direct hypernym of each other, iii) one word is the indirect hypernym of the other and iv) adjectives have a similar relation with each other.

Wu et al. (2013a) explore different types of semantic representations such as: named entities, distributional semantics and structured distributional semantics. The method combines one of the state-of-the-art methods (Bär et al., 2012) with features extracted from the semantic representations. The semantic features complement the state-of-the-art features by using a feature selection algorithm. The structured distributional metric improves the performance of the state-of-the-art features by incorporating syntactic information into the distributional model.

Noeman (2013) propose a method based on the combination of different types of similarity metrics. The main contribution is towards showing the effect of metrics on STS such as: lexical matching, lexical matching with term frequency and inverse document frequency, modified BLEU and named

entities matching. The inverse document frequency lexical matching leverage information from a corpus to expand the standard lexical matching. The modification of BLEU relies on the alignment of exact words, stems and synonyms. The final similarity metric is a linear combination of features, where the weights are tuned manually. The best combination of features is the idf lexical matching with the lexical matching using stemmed words and synonym matching.

Shareghi and Bergler (2013) present a method based on an exhaustive combination of 11 simple lexical features. The method uses a Support Vector Regressor with all possible combinations of the features and it trains separate models based on each combination. The model creates a meta-feature space and it trains a final model based on it. The two-step method outperforms the one using individual models.

2.2.2 Task Adaptation

The motivation behind this model is that task adaptation techniques will alleviate the generalisation problem imposed by unknown tasks. The methods leverage information for the STS training datasets to improve performance on unknown test tasks. In addition, these types of methods also use similarity metrics to train regression algorithms.

Heilman and Madnani (2013) propose to incorporate domain/task adaptation techniques (Daumé et al., 2010) for STS to generalise models to new tasks. Heilman and Madnani (2013) add extra features into the model, where

the feature set contains task specific features plus general task (i.e. unknown task) features. The ML algorithm infers the extra weights for each specific task and for the general task. When an instance of a specific task is to be predicted, only the copy of the features of that task will be active; if the task is unknown, the general features will be active.

Severyn et al. (2013) propose to use meta-classification to cope with task adaptation. The model merges each text pair into one text and extracts meta-features from them. The meta-classification model predicts, for each instance, its most likely task based on the previous features. When an instance is classified into its mots likely task, the model uses features based on tree kernels that automatically extract syntactic information to predict similarity.

In sum, methods for STS use a wide variety of features to train regression algorithms. The similarity metrics use different sources of information such as: tokenisation, lemmatisation/stemming, Named Entity Recognition, part-of-speech tagging or dependency parsing.

Chapter 3

Methods for Measuring the Directional Relation of Texts

In this Chapter, we describe our proposed method for RTE based on a multi-stage architecture model. We divide our architecture into two stages: i) alignment stage and ii) entailment decision stage. For the first stage we propose different alignment methods based on the matching of predicate-argument structures. In the second stage we use different learning schemes based on information from the alignment stage to predict the entailment relation.

3.1 Alignment Stage

We have focused the alignment stage on matching predicate-argument structures between a T-H pair. Our assumption is that the information about similar situations with similar participants between a T-H pair can be used as evidence to decide on entailment. Previous work that uses the semantic role matching is based on exact matching of roles and role fillers such as in (Giménez et al., 2010) and (Burchardt and Frank, 2006). However, exact matching is a limitation and it is not clear what the contribution of this specific information is for the overall performance of their systems. Thus,

the goals of our alignment stage are: i) comparing both the semantic structure and its content across matching arguments in the hypothesis H and text T; and ii) using alternative ways of measuring inexact matches for both predicates and role fillers.

The alignment uses SRL annotation to define a predicate-argument structure. However, it analyses the content of predicates and arguments seeking for either exact or similar matches. For example, the following T-H pair:

T The lack of snow is putting people off booking ski holidays in hotels and guest houses.

H The lack of snow discourages people from ordering ski stays in hotels and boarding houses.

Each predicate in T is tagged with the corresponding frames:

book The lack of snow is putting [people]$_{A0}$ off [booking]$_{V}$ [ski holidays]$_{A1}$ in [hotels and guest houses]$_{AM-LOC}$.

put [The lack of snow]$_{A0}$ is [putting]$_{V}$ [people]$_{A1}$ [off booking ski holidays in hotels and guest houses]$_{A}$2.

The same applies for H:

discourage [The lack of snow]$_{A0}$ [discourages]$_{V}$ [people]$_{A1}$ [from ordering ski stays in hotels and boarding houses]$_{A2}$.

order The lack of snow discourages [people]$_{A0}$ from [ordering]$_{V}$ [ski stays]$_{A1}$ in [hotels and boarding houses]$_{AM-LOC}$.

In this work, we show one pair of frames (e.g. predicates book and discourage) for each example of alignment between a T-H pair. We use the SENNA[1] parser (Collobert et al., 2011) as source of SRL annotation. SENNA have achieved an state-of-the-art performance with an F-measure of 75.79% for tagging semantic roles over the CoNLL 2005 [2] benchmark.

The meaning of the Argument tags is as follows:

Arguments:

- A0 subject
- A1 object
- A2 indirect object

Adjuncts:

- AM-ADV adverbial modification
- AM-DIR direction
- AM-DIS discourse marker
- AM-EXT extent
- AM-LOC location
- AM-MNR manner
- AM-MOD general modification

[1]The SENNA parser v2.0 outputs the numbers as 0
[2]http://www.lsi.upc.edu/ srlconll/

- AM-NEG negation
- AM-PRD secondary predicate
- AM-PRP purpose
- AM-REC recipricol
- AM-TMP temporal

Other Labels:

- C-arg continuity of an argument/adjunct of type arg
- R-arg reference to an actual argument/adjunct of type arg

We propose three different alignment methods:

TINE Lexical Matching based on the inexact matching of predicate-argument structures with ontologies and distributional semantics.

TINE Context Matching based on an optimisation step to align predicates given the context (i.e arguments).

TINE Edit Distance based on a modification of the edit distance method to allow the use of predicate-argument information. This method matches structures, and then transforms T into H to compute the distance between them.

3.1.1 TINE Lexical Matching

In (Rios et al., 2011), we propose an alignment method that complements lexical matching with a shallow semantic component. The main contribution is to provide a more flexible way of measuring the overlap between shallow semantic representations that considers both the semantic structure of the sentence and the content of the semantic elements. The inexact matching is based on the use of ontologies such as VerbNet (Schuler, 2006) and distributional semantics similarity metrics, such as Dekang Lin's thesaurus (Lin, 1998b). This is an automatically built thesaurus, and for each word it has an entry with the most similar words and their similarity scores.

$$A(H,T) = \frac{\sum_{v \in V} verb\_score(H_v, T_v)}{|V_t|} \tag{3.1}$$

In Equation 3.1, V is the set of verbs aligned between H and T, and $|V_t|$ is the number of verbs in T. Hereafter the indexes h and t stand for hypothesis and text, respectively. Verbs are aligned using VerbNet (Schuler, 2006) and VerbOcean (Chklovski and Pantel, 2004). A verb in the hypothesis v_h is aligned to a verb in the text v_t if they are related according to the following heuristics: (i) the pair of verbs shares 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*.

$$verb\_score(H_v, T_v) = \frac{\sum_{a \in A_t \cap A_h} arg\_score(H_a, T_a)}{|A_t|} \tag{3.2}$$

The similarity between the arguments of a verb pair (v_h, v_t) in V is measured as defined in Equation 3.2, where A_h and A_t are the sets of labeled arguments of the hypothesis and the text respectively and $|A_t|$ is the number of arguments of the verb in T. In other words, we only measure the similarity of arguments in a pair of sentences that are annotated with the same role. This ensures that the structure of the sentence is taken into account (for example, an argument in the role of *agent* would not be compared against an argument in a role of *experiencer*). Additionally, by restricting the comparison to arguments of a given verb pair, we avoid argument confusion in sentences with multiple verbs.

The $arg\_score(H_a, T_a)$ computation is based on the cosine similarity for BoW. We treat the tokens in the argument as a BoW. However, in this case we change the representation of the segments. If the two sets do not match exactly, we expand both of them by adding similar words. For every mismatch in a segment, we retrieve the 20-most similar words from Dekang Lin's distributional thesaurus (Lin, 1998b), resulting in sets with richer lexical variety.

The following example shows how the computation of $A(H, T)$ is performed, considering the following example:

T The lack of snow is putting [people]$_{A0}$ off [booking]$_{V}$ [ski holidays]$_{A1}$ in [hotels and guest houses]$_{AM-LOC}$.

H The lack of snow discourages [people]$_{A0}$ from [ordering]$_{V}$ [ski stays]$_{A1}$ in [hotels and boarding houses]$_{AM-LOC}$.

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}

   argument_score = 1

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...}}

argument_score(H_{A1}, T_{A1}) = cosine (H_{A1}, T_{A1})

argument_score(H_{A1}, T_{A1}) = $\frac{|H_{A1} \bigcap T_{A1}|}{\sqrt{|H_{A1}| \times |T_{A1}|}}$

argument_score = 0.5

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

   argument_score(H_{AM-LOC}, T_{AM-LOC}) = cosine (H_{AM-LOC}, T_{AM-LOC})

   argument_score = 0.72

9. verb_score (*order*, *book*) = $\frac{1+0.5+0.72}{3}$ = 0.74

10. $A(H,T) = \frac{0.74}{2} = 0.37$

Different from previous work, we have not used WordNet to measure lexical similarity for two main reasons: problems with lexical ambiguity and limited coverage in WordNet (instances of named entities are not in WordNet, e.g. *Barack Obama*).

For example, in WordNet the aligned verbs (*order/book*) from the previous examples have: 9 senses - *order* (e.g. give instructions to or direct somebody to do something with authority, make a request for something, etc.) - and 4 senses - *book* (engage for a performance, arrange for and reserve (something for someone else) in advance, etc.). Thus, a WordNet-based similarity measure would require disambiguating segments, an additional step and a possible source of errors. Second, a threshold would need to be set to determine when a pair of verbs is aligned. In contrast, the structure of VerbNet (i.e. clusters of verbs) allows a binary decision.

In summary, this method outputs a similarity score between two predicate-argument structures, and an alignment matrix where the predicates and arguments are related with a degree of similarity. This output is given the learning algorithm.

3.1.2 TINE Context Matching

In Rios et al. (2012), we propose modifications to the previous alignment method, where the matching of unrelated verbs is a crucial issue, since the sentences to be compared are not necessarily very similar. We have thus modified TINE Lexical Matching with an optimisation step which aligns the verb predicates by measuring two degrees of similarity: i) how similar their arguments (context) are, and ii) how related the predicates' realisations are. Both scores are combined as shown in Equation 3.3 to score the similarity between the two predicates (H_v, T_v) from a pair (T, H).

$$\begin{aligned} \mathrm{sim}(\mathrm{H_v}, \mathrm{T_v}) = (w_{lex} \times lexScore(H_v, T_v)) \\ +(w_{arg} \times argScore(H_{arg}, T_{arg})) \end{aligned} \tag{3.3}$$

where w_{lex} and w_{arg} are the weights for each component, $argScore(H_{arg}, T_{arg})$ is the similarity, which is computed as the BoW cosine similarity of the arguments between the predicates being compared. Again, we treat the tokens in the argument as a BoW. $lexScore(H_v, T_v)$ is the similarity score extracted from the Dekang Lin's thesaurus between the predicates being compared. If the verbs are related in the thesaurus we use their similarity score as

$lexScore$, otherwise $lexScore = 0$. The pair of predicates with the maximum sim score is aligned. The alignment is an optimisation problem where predicates are aligned 1-1: we search for all 1-1 alignments that lead to the maximum average sim for the pair of sentences. For example,

T The tech - loaded [Nasdaq composite]$_{A1}$ [rose]$_{V}$ [0 points]$_{A2}$ [to 0]$_{A2}$, [ending at its highest level for 0 months]$_{AM-ADV}$.

H The technology - laced [Nasdaq Composite Index]$_{A1}$ IXIC [climbed]$_{V}$ [0 points , or 0 percent ,]$_{A2}$ [to 0]$_{A4}$.

have the following list of predicates: T = {loaded, rose, ending} and H = {laced, climbed}. The method compares each pair of predicates and aligns the predicates *rose* and *climbed* because they are related in the thesaurus with a similarity score $lexScore = 0.796$ and a $argScore = 0.185$ given that the weights are set to 0.5 and sum up to 1, the predicates reach the maximum sim = 0.429 score. The output of this method results in a set of aligned verbs between a pair of sentences. As in the previous method, the alignment matrix can be used to compute a similarity metric as well.

3.1.3 TINE Edit Distance

In (Rios and Gelbukh, 2012b), we propose a semantic edit distance metric to address the limitations of the TINE Lexical Matching method when is used as a similarity score feature. The goal of our semantic edit distance metric is to measure the differences between a T-H pair at a predicate-argument level.

This metric is based on the same assumption as the transformation methods for RTE, where the entailment relations can be measured by applying a series of transformations from T into H.

The semantic edit distance metric is a modified version of the TINE Lexical Matching method. Thus, instead of outputting an alignment matrix this method will compute a distance score between the T-H pair. The modified version of the alignment method is divided into two stages: i) The automatic alignment of predicates and ii) The edition of the arguments between the aligned predicates.

In the alignment stage a set of Verbs between the T-H pair are aligned using VerbNet and VerbOcean as in TINE Lexical Matching.

For the edition stage we define three operations over arguments:

1. Deletion of an argument

2. Insertion of an argument

3. Substitution of an argument

The Deletion operation is applied if one of the arguments in H is missing in T, the Insertion operation is applied if one of the arguments in T is missing in H, and the Substitution operation is applied if the arguments in the T-H pair are of the same type but they have a different word realisation. For each pair of verbs from the set supplied by the previous stage, Equation 3.4 is computed. The final score is the average score over the total of verbs Equation 3.5.

$$edition\_score(T_v, H_v) = \frac{1}{number\_of\_operations}, \tag{3.4}$$

where T_v is an aligned verb in the Text, H_v is the corresponding aligned verb in the Hypothesis, and $number\_of\_operations$ is the addition of the applied operations for each argument on T. In this version the metric does not give a score weight to each operation.

$$semantic\_score(T, H) = \frac{1}{n}\sum_{v=1}^{n} edition\_score(T_v, H_v), \tag{3.5}$$

where n is the number of verbs.

The following example shows how the computation of $edition\_score(T_v, H_v)$ is performed:

T Recent Dakosaurus research comes from a complete skull found in Argentina in 0, $[\text{studied}]_V$ [by Diego Pol of Ohio State University, Zulma Gasparini of Argentinas National University of La Plata, and their colleagues]$_{A0}$.

H $[\text{A complete Dakosaurus}]_{A1}$ was $[\text{discovered}]_V$ $[\text{by Diego Pol}]_{A0}$.

1. extract verbs from T: T_v = {comes,found, studied}
2. extract verbs from H: H_v = {discovered}
3. similar verbs aligned with VerbNet (shared class discover-84-1-1) V = {(study, discover), (find, discover)}
4. Apply operations over T arguments for (study, discover):
   operation 1:insert A1 = {A complete Dakosaurus}

operation 2:substitution A0 ={by Diego Pol}

$edition\_score(study, discover) = 1/2 = 0.5$

The transformed T for the pair of predicates (study and discover) is:

[A complete Dakosaurus]$_{A1}$ [studied]$_{V}$ [by Diego Pol]$_{A0}$.

In terms of implementation we store each chunk into a stack this means that the order of chunks in the sentence is lost. The edition score is not based on the order of the chunks, but in the number of applied operations. The order can be tracked given the positions of the chunks (i.e. arguments) in the H frame.

However, the alignment stage may not be able to match any verb. We use a back-off score metric in case the score of the edit distance is zero. The back-off score is based on shallow syntactic annotation or Chunking Eq. (3.10). This method gives as output a score given the series of transformations from T to H. We then use the output of this metric only for the entailment decision.

3.2 Entailment Decision Stage

In this section we describe the different learning models used to predict the entailment decision.

3.2.1 Propositional Learning Model

We use a supervised ML algorithm to decide the entailment relation. The RTE 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 (Dagan et al., 2010).

As a first attempt to solve RTE, we use the previously described alignment methods as features in a standard ML-based method. In addition, we propose new similarity metrics based on different representations of text for RTE that are: i) Chunks, and ii) Named entities (Rios and Gelbukh, 2012a). The goal of these new features is to address the previously discussed limitations of existing semantic-based alignment methods.

The similarity metrics are a fundamental part of the similarity models for RTE. The motivation to propose different types of similarity metrics is to exploit their complementaries: if one metric is not able capture (identify) an RTE phenomenon, a different type of metric will be able to capture the missing information. Moreover, the new metrics are different from previous work by allowing the matching (alignment) of similar content of the semantic units, whereas the standard metrics only match exact content. For example, in the named entity metric, the entities of similar type are grouped but the metric allows synonym surface realisations.

In order to train a classifier we extracted features based on the scores of similarity metrics. The ML model is divided in tree steps: i) pre-process data, ii) train a classifier and iii) classification. The similarity metrics are: token-based similarity metrics, a syntactic similarity metric, a named entity similarity metric and semantic similarity metric (e.g. a given predicate-argument alignment).

Token features The token-based features are as follows: Cosine similarity (Equation 3.6), Precision (Equation 3.7), Recall (Equation 3.8), and F-measure (Equation3.9).

$$cosine(T, H) = \frac{|T \bigcap H|}{\sqrt{|T| \times |H|}} \tag{3.6}$$

$$precision(T, H) = \frac{|T \bigcap H|}{|H|} \tag{3.7}$$

$$recall(T, H) = \frac{|T \bigcap H|}{|T|} \tag{3.8}$$

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.9}$$

Where T and H are a bag-of-token representation of the T–H pair. We use different types of token representations such as: i) word, ii) lemma, and iii) Part-of-Speech (POS). We compute the similarity between the T-H pairs for each token representation and metric. These scores become features for the classifier.

Chunking feature The motivation of a chunking similarity metric is that a T-H pair with a similar syntax can hold an entailment relation. Shallow parsing is a partial syntactic representation of texts. It is an alternative to full parsing because it is more efficient and 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 group).

The chunking feature is defined as the average of the number of similar chunks (in the same order) between the T-H pairs.

$$chunking(T, H) = \frac{1}{m} \sum_{n=1}^{m} simChunk(t_n, h_n), \tag{3.10}$$

where m is the number of chunks in T, h_n is the n chunk tag and content in the same order, $simChunk(t_n, h_n) = 1$ 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 $simChunk(t_n, h_n)$ 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 3.2.1 shows an example of how the metric computes the partial

scores. Finally, the metric (Equation 3.10) computes the overall score $chunking(T, H) = 0.64$.

Table 3.1: Example of $simChunk(t_n, h_n)$ partial scores

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

Named entities feature Named Entity Recognition (NER) is a task which identifies and classifies parts of a text that represent entities into predefined classes such as names of persons, organisations, 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 Organisation.

The motivation of a similarity metric 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 measure also deals with synonym entities.

Our method for NER similarity measure consists in the following: First, the named entities are grouped by type (e.g., Scripps Hospital is an Or-

ganisation) and then the content of the same type of groups is compared using cosine similarity. If the surface realisations are different, we retrieve words that share the same context as the named entity (using Dekang Lin's thesaurus). The cosine similarity thus takes into account more information than just the named entities.

$$ne\_score(T, H) = \frac{1}{m} \sum_{n=1}^{m} simNER(t_n, h_n), \tag{3.11}$$

where m is the number of named entities, h_n is the n entity tag and content. In $simNER(t_n, h_n) = 0$ if the tags are different $t_n(TAG) \neq h_n(TAG)$, otherwise $simNER(t_n, h_n) = cosine(t_{entity}, h_{entity})$. The t_{entity} and h_{entity} are BoW's extracted from the thesaurus.

For example, the $simNER$ score for one named entity from the T-H pair:

T Along with chipmaker Intel , the companies include Sony Corp. , Microsoft Corp. , NNP Co. , [IBM Corp.]$_{ORG}$, Gateway Inc. and Nokia Corp.

H Along with chip maker Intel , the companies include Sony , Microsoft , NNP , [International Business Machines]$_{ORG}$, Gateway , Nokia and others.

1. group entities from similar tag: ORG (i.e. Organisation)

   T: *IBM Corp.*

H: *International Business Machines*

2. add words from the similarity thesaurus resulting in the following BoW's:

   T_{entity}: {IBM Corp.,... Microsoft, Intel, Sun Microsystems, Motorola/Motorola, Hewlett-Packard/Hewlett-Packard, Novell, Apple Computer...}

   H_{entity}:{International Business Machines,... Apple Computer, Yahoo, Microsoft, Alcoa...}

3. $simNER(T_{entity}, H_{entity}) = cosine(T_{entity}, H_{entity})$

   $simNER(T_{entity}, H_{entity}) = 0.53$

With the previous metrics Eqs. (3.6) to (3.11) and the similarity output from a given alignment method, we build a vector of similarity scores used as features to train an ML algorithm. We use the RTE Challenges datasets to train and test the following ML algorithms: SVM, NaïveBayes, AdaBoost, BayesNet, LogitBoost, MultiBoostAB, RBFNetwork, and VotedPerceptron. We follow a standard experimental design for ML on RTE (Malakasiotis and Androutsopoulos, 2007). We use the default configuration of the ML algorithms. First, for the training step we conduct a 10 fold-cross validation test in order to chose the algorithm with the best performance over the training dataset. Finally, we use the best ML algorithm for testing.

3.2.2 Statistical Relational Learning Model

In (Rios et al., 2014), we propose the use of a Statistical Relational Learning model for RTE. Instead of using simple similarity metrics to predict the entailment decision, we use rich relational features extracted from output of the predicate-argument alignment structures between T-H pairs. These are used to train an MLN framework, which learns a model to reward pairs with similar predicates and similar arguments and penalise pairs otherwise.

Different from (Garrette et al., 2011), 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.

In the entailment decision stage we use an MLN to predict the entailment relation of a given T-H pair. As an inherently semantic task, RTE should naturally benefit from knowledge about the relationships among elements 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).

MLN (Richardson and Domingos, 2006) 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.

A first-order logic knowledge base (KB) is a set of sentences or formulas in first-order logic. The formulas are built using four types of symbols:

1. Constant symbols, which represent objects in the domain of interest. For example, people: Anna, Bob, Chris.

2. Variable symbols, which range over the objects in the domain. For example, x.

3. Function symbols, which represent mappings from tuples of objects to objects. For example, $Friends$.

4. Predicate symbols, which represent relations among objects in the domain. For example, $Friends(x, y)$.

An atomic formula or atom is a predicate symbol applied to a tuple of terms. For example, $Friends(x, y)$. The formulas are built in a recursive way using logical connectives and quantifiers with the four types of symbols. Lets say that $F1$ and $F2$ are formulas. The following are formulas as well:

negation $\neg F1$, conjunction $F1 \wedge F2$ and implication $F1 \Rightarrow F2$, among others. A possible world assigns a truth value to each possible ground atom (i.e. terms containing only constants, $Friends(Anna, Bob)$).

The first-order logic is a set of hard constraints on the set of possible worlds that define our domain. If a world violates one formula, the probability of this world is zero. However, MLN extends first-order logic by softening this constraint: when a world violates a formula of the KB, it becomes less probable, but will not have a zero probability. A world with fewer violated formulas is more probable. Therefore, each formula is associated with a weight learnt from data which defines how strong this constraint is.

MLN can be seen as a template for constructing Markov networks. A Markov network is a model for the joint distribution of a set of variables $X = (x_1, x_2, ..., x_j)$. It is composed by an undirected graph G and a set of potential functions. Markov networks are often conveniently represented as *log-linear models*, a potential function for each clique is replaced by an exponentiated weighted sum of features given by:

$$P(X = x) = \frac{1}{Z} exp(\sum_j w_j f_j(x)),$$

where the MLN is the knowledge base composed by the pairs (w_j, f_j) where each f_j is a formula (i.e. potential function) in first-order logic and w_j is a weight. The partition function Z assigns the function scores into a probability. The learning problem consists thus in tuning the weights to match a

distribution of possible worlds which is consistent with the real world (i.e., the training data).

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.

The following aligned T-H pair with $id = 1$ will help us to explain the ML rules notation:

T [Russian cosmonaut Valery Polyakov]$_{A0}$ [set]$_{V}$ [the record for the longest continuous amount of time spent in space, a staggering 0 days, between 0 and 0]$_{A1}$.

H [Russians]$_{A0}$ hold [record for longest stay in space]$_{A1}$.

The entailment relation for the pair $id = 1$ is *TRUE*. The alignment method finds the predicates *set* and *hold* equivalent given their context (i.e. arguments). The alignment of arguments is as follows:

A0 [Russian cosmonaut Valery Polyakov] $\Leftrightarrow$ [Russians]

A1 [the record for the longest continuous amount of time spent in space, a staggering 0 days, between 0 and 0] $\Leftrightarrow$ [record for longest stay in space]

From the previous alignment structure we extract different information to build the predicates that constitute MLN rules. Our baseline using an MLN model is based on simple features like rules to model the entailment decision. The simple non-relational rules are as follows:

Bag-of-words and Part of Speech (PoS) tags For each token in the T-H pair we extract their lemmas and part of speech tags. We represent it by the predicate $TokenBaseline(pid, token)$. For example, the token *cosmonaut* produces the following predicates $TokenBaseline(1, cosmonaut)$ and $TokenBaseline(1, NN)$.

Word Overlap For each T-H pair we compute the number of lemmas shared between the T and H: $Overlap(pid, num)$. For example, the number of shared words for pair id 1 is 6 $Overlap(1, 6)$.

pid is the id of a T-H pair, *token* is the lemma or the PoS tag (each one has a separate predicate), and *num* is the overlap score.

We define the following MLN formulas for the entailment decision:

$$\begin{aligned} &TokenBaseline(pid, +token) \\ &\qquad \Rightarrow Entailment(+d, pid) \\ &Overlap(pid, +n) \\ &\qquad \Rightarrow Entailment(+d, pid) \end{aligned} \tag{3.12}$$

An example of grounding for the previous MLN rule is:

$TokenBaseline(1, cosmonaut) \Rightarrow Entailment(TRUE, 1)$

Our Relational 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 wrong or poor information about the relations between the participants and situations of the entailment 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 Context Matching.

The following variables are created to represent this information: Arg and $Verb$. Figure 3.1 shows the relationships between these variables in a Markov Network.

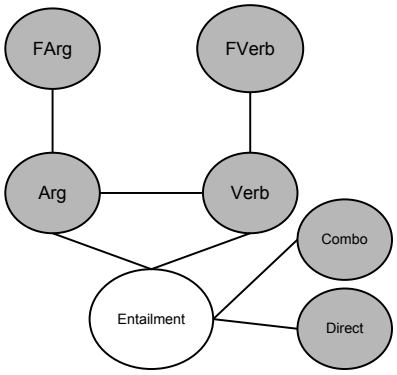

Figure 3.1: Markov network of our RTE model

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 related to them: $FArg$ is the set of features related to the arguments, and $FVerb$ is the set of features related to the verbs.

The following are features for each token of aligned arguments:

Lexical Word, lemma and PoS of each token. For example, the predicate $Token(A0, 1, cosmonaut)$.

Similar Words The 20 most similar words from Dekang Lin's thesaurus for each token. A predicate is created for each similar word. For example, the predicate $Token(A0, 1, astronaut)$.

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. For example, the predicate $Token(A0, 1, spaceman)$.

These argument features are represented by the following formula:

$$Token(aid, pid, +tfeature) \wedge Arg(aid, vid, pid) \Rightarrow Entailment(+d, pid) \quad (3.13)$$

where $tfeature$ takes the value of each of the previous features, aid and vid are the values of the Arg and $Verb$ variables. For example, the rule grounding given the aligned verbs (set and hold) and the token predicate is

a hypermyn:

Token(A0, 1, spaceman) $\wedge$ Arg(A0, set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

For the aligned verbs, we extract:

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:

$$BowVN(vid, +bowfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid) \tag{3.14}$$

For example, the rule for the aligned verbs (set and hold) is:

BowVN(set-hold, "fit-54.3 put-9.1.2 hold-15.1.1") $\wedge$ Verb(set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

Strong Context $strfeature$ compares components in Eq. 3.3. 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:

$$StrongCon(vid, +strfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid) \tag{3.15}$$

For example, the rule for the aligned verbs (set and hold) is:

StrongCon(set-hold, 1) $\wedge$ Verb(set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

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

$$SimVN(vid, +simvnfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid) \tag{3.16}$$

For example, the rule for the aligned verbs (set and hold) is:

SimVN(set-hold, 0) $\wedge$ Verb(set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

Similarity VerbOcean $simvofeature$ is 1 if the verbs have the *similar* relation as given by VerbOcean (Chklovski and Pantel, 2004);[3] 0 otherwise:

$$SimVO(vid, +simvofeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid) \tag{3.17}$$

For example, the rule for the aligned verbs (set and hold) is:

SimVO(set-hold, 0) $\wedge$ Verb(set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

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

$$TokenVerb(vid, +tokenvfeature) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid) \tag{3.18}$$

For example, the rule for the aligned verbs (set and hold) is:

TokenVerb(set-hold, "set hold") $\wedge$ Verb(set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

[3]VerbOcean contains different relations between verbs.

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

$$Arg(aid, vid, pid) \wedge Verb(vid, pid) \Rightarrow Entailment(+d, pid) \tag{3.19}$$

For example, the rule for the aligned verbs (set and hold) and the argument A0 is:

Arg(A0, set-hold, 1) $\wedge$ Verb(set-hold, 1) $\Rightarrow$ Entailment(TRUE, 1).

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, it is possible that the metric cannot align some of the T-H pairs, mostly because SENNA does not produce any SRL structure for certain pairs. In order 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 3.1. *Combo* holds the value $cfeature$ which consists of all the combinations of unigrams between the H-T pair. We create a new predicate for each unigram combination:

$$Combo(pid, +cfeature) \Rightarrow Entailment(+d, pid) \tag{3.20}$$

For example, the rule from the combinations of tokens (cosmonaut and space) is: Combo(1, "cosmonaut space") ⇒ Entailment(TRUE, 1).

The *Direct* variable holds the value $simdfeature$ with 1 if the verbs hold an entailment relation as given by the Directional Database (Kotlerman et al., 2010); 0 otherwise:

$$Direct(pid, +simdfeature) \Rightarrow Entailment(+d, pid) \tag{3.21}$$

The database contains directional lexical entailment rules. The lexical entailment rules are, for example, koala ⇒ animal, bread ⇒ food. The meaning of the left-hand-side implies the meaning of its right-hand-side. For example, the rule from the pair of tokens (cosmonaut and space) is: Direct(1, 0) ⇒ Entailment(TRUE, 1)

The Markov Network built from these formulas we can 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.

For the MLN models we use the Alchemy[4] toolkit and the datasets from the RTE challenges 1-3 (Dagan and Glickman, 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007), which are publicly available, to evaluate our MLN models. To predict the entailment decision we take the marginal probabilities

[4] http://alchemy.cs.washington.edu/

that Alchemy outputs for a given query, i.e. the *Entailment* predicate. The query with the highest probability gives the entailment decision.

3.3 Recognising Textual Entailment Evaluation Datasets

The RTE datasets consist of small text snippet's pairs manually labelled for entailment, corresponding to the news domain. The datasets provided by the RTE challenge organisers are intended to include text pairs corresponding to success and failure results of NLP applications such as: IE, IR, QA and SUM. The datasets are divided into two: development and test datasets. In addition, Dagan and Glickman (2005) define the official guidelines for the RTE as follows:

- Entailment is a directional relation; the hypothesis must be entailed by the text and not the contrary.
- The hypothesis must be fully entailed by the text and do not include parts which could not be inferred.
- Cases in which the inference is likely to be high but not with absolute certainty, should be judge as true.
- The background knowledge about the world must be typical to a normal reader of that kind of text (news domain); it is not acceptable the known presupposition of high specific knowledge.

The definition of RTE is based on two assumptions: the common human understanding of language as well as common background knowledge. An example of background knowledge is: a company has a CEO, a CEO is an employee of the company, an employee is a person and so on.

The judgements (classifications) produced by the methods are compared to a gold standard, where the accuracy and the average precision are used to evaluate the performance of each method. The principal measure used to evaluate this task is *accuracy*, which is the percentage of correctly classified decisions. The second evaluation measure is the confidence weighted score (also known as Average Precision), in which the judgements of the test examples are sorted by their confidence (in decreasing order).

Several NLP applications may benefit from RTE. For example:

- In Summarisation, a summary should be entailed by the source text (Lloret et al., 2008).
- In Information Extraction, the information extracted by the system should also be entailed by the source text (Androutsopoulos and Malakasiotis, 2010).
- In Question Answering the answer, which is obtained from the question after an IR process, must be entailed by a supporting snippet of text (Negri and Kouylekov, 2009).
- In MT Evaluation, the meaning of the machine translation should entail the meaning of a human reference translation (Padó et al., 2009b).

- In Paraphrase Recognition (PP), there is a mutual entailment between the T–H pair (Rus et al., 2008).

- In coreference resolution, the longest most informative mention in the coreference chain across the T-H pair serves as evidence to decide the entailment relation (Mitkov et al., 2012).

The first RTE challenge (Dagan and Glickman, 2005) started in 2005, and it was organised by Ido Dagan, Oren Glickman and Bernardo Magnini. The main goal was to develop a framework to evaluate the performance of the participating methods over the RTE task. In total 17 groups participated in the challenge which showed that the RTE task is relevant for various applications.

The datasets come from different text processing applications such as: IR, QA, IE and MT. The examples represent a range of different levels of entailment such as: reasoning, lexical, syntactic, logical and world knowledge. A total of 567 entailment pairs are part of the development dataset and 800 pairs are part of the test dataset, the datasets are split into TRUE/FALSE examples. For the manual evaluation each T-H pair is judged by a first annotator, and the pairs are cross-evaluated by a second judge, who received only the pair without any additional information from the original context. The annotators agreed in their judgement for 80% of the pairs, reaching a 0.6 Kappa level (moderate agreement). The remainder 20% of the pairs with disagreement among the judges were discarded of the final dataset.

The main goal of the second RTE-2 Challenge (Bar-Haim et al., 2006) was to provide more "realistic" entailment pairs. The datasets have 1600 pairs divided into development and test, each one containing 800 pairs. The challenge target was four applications: IR, IE, QA and SUM. As in the previous challenge most of the approaches were based on ML algorithms. The agreement of the dataset was computed in the same way as the RTE-1. The annotators agreed in 89.2% of the pairs with an average Kappa level of 0.78, which corresponds to substantial agreement. Where 18.2% of the pairs were discarded because of disagreement.

The RTE-3 Challenge (Giampiccolo et al., 2007) followed the same structure of the previous versions. However, a resource pool was introduced, where participants had the possibility to share the same resources. For the manual annotation the annotators agreed on 87.8% of the pairs, with an average Kappa level of 0.75 (substantial agreement), and 19.2% of the pairs were discarded because of disagreement.

The RTE-4 Challenge (Giampiccolo et al., 2008) introduced the three-way decision of "ENTAILMENT", "CONTRADICTION" and "UNKNOWN", where methods have to make more precise decisions. For example, a hypothesis is unknown if there is not enough evidence to support the entailment decision. Also, the text should be distinguished from a hypothesis to be false or contradicted by that text. The two-way RTE task was still a part of the challenge, in which the pairs where T entailed H were marked as ENTAIL-

MENT, and those where the entailment did not hold were marked as NO ENTAILMENT. The three-way task guidelines are as follows:

- T entails H - in which case the pair will be marked as ENTAILMENT.
- T contradicts H - in which case the pair will be marked as CONTRADICTION.
- The truth of H cannot be determined on the basis of T - in which case the pair will be marked as UNKNOWN.

The RTE-4 dataset contains 1000 pairs (300 pairs each from IE and IR, 200 pairs each from SUM and QA). The annotator agreement was not published in this Challenge, but the same method of discarding pairs as in previous Challenges was used. From RTE-4 onwards, the datasets are not publicly available, only upon request.

The impact of discourse information is measure in the context of the RTE Search Task [5]. The RTE Search Task consist in identifying all the sentences among candidate sentences, which entail a given Hypothesis. In other words, given a corpus and a set of "candidate" sentences (documents) retrieved by an IR engine from that corpus the systems decide if the candidate sentences entail the hypothesis.

[5] http://www.nist.gov/tac/2011/RTE/

3.4 Results and Discussion

In this Section, we test our proposed models with different RTE datasets. We discuss and compare the results of our proposed models with previous work. We define the following names for our proposed models:

ML-TINE: SVM classifier using the features of the Propositional model Eqs. (3.6) to (3.11) and TINE Context Matching Eq. (3.1). We use the SVM implementation with the default configuration from scikit-learn (Pedregosa et al., 2011).

ML-EDIT: SVM classifier using the features of the Propositional model Eqs. (3.6) to (3.10) and TINE Edit Distance Eq. (3.5). We use the SVM implementation with the default configuration from scikit-learn (Pedregosa et al., 2011).

MLN-BASE: MLN model using the simple rules from Eq. (3.12). We use the Alchemy toolkit with the default configuration.

MLN-RELATIONAL: MLN model using relational rules from Eqs. (3.13) to (3.21). We use the Alchemy toolkit with the default configuration.

For the alignment stage we use the TINE Context Matching method given that TINE Lexical Matching method makes simple errors such as the matching of unrelated verbs and suffers from the lack of coverage of the ontologies. For example, in the following T-H pair:

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.*

The verbs *remain* and *say* are (incorrectly) related as given by the VerbOcean heuristic. Then, TINE Context Matching method is a more robust source of alignment information.

We compare ML-TINE with other ML-based methods, and with methods that use a SRL representation as features. In addition, we train a baseline system using a SVM classifier. This baseline is based on simple representations of the T-H pairs and simple string similarity metrics. The goal is to find the best combination of simple representations/features that optimise accuracy over the test datasets to train a classifier. The motivation is to compare a strong baseline based on simple features with our proposed methods. We also compare our method with the baseline proposed by Mehdad and Magnini (2009) as mentioned in Chapter 2. We defined the previous system as *official baseline*.

The representations are: tokens, lemmas and PoS, which are extracted from the TreeTagger[6]. The string similarity metrics are: word overlap, cosine, dice, jaccard, and overlap. These metrics are based on set operations over BoW's. For example, the cosine metric is Eq. 3.6. The full description of

[6]http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

these metrics is described in (Manning and Schütze, 1999). We define this baseline system as *SVM-gen*.

The Baseline system uses the following features:

1. Tokens with Word overlap, Cosine, Dice, Jaccard, and Overlap
2. Lemmas with Word Overlap, Cosine, Dice, Jaccard, and Overlap
3. PoS with Word overlap, Cosine, Dice, Jaccard, and Overlap

The result is a vector of 15 features where each representation of the T-H pair is scored by a different lexical metric. For example, with the previous 15 features over the RTE-3 development dataset. Using an SVM with 10-fold cross-validation results in an accuracy of 63.25%. The design of the genetic algorithm is as follows:

- The chromosome of the individual is composed of three genes. Each gene is one of the representations (i.e. token, lemma and PoS), and each representation could be measure with one of the similarity metric (i.e. word overlap, cosine, dice...)
- The fitness function is the accuracy of the features over a given development set
- A population of 80 individuals
- Number of generations 1,000
- A crossover probability of 60%

- A mutation probability of 10%
- The selection method is the Roulette scheme
- The crossover strategy is two point
- The offspring comes from two parents
- The tree best individual of the previous generation are preserved

The standard parameter setting for a genetic algorithm is (Jong and Spears, 1990) individual size 50, number of generations 1,000, crossover type typically two point, crossover rate of 60% and mutation rate of 10%. We increase the number of individuals given that our fitness function is fairly fast to compute (i.e. the standard RTE dataset has 800 instances).

The features chosen by the genetic algorithm feature selection are:

1. Tokens with Overlap
2. Lemmas with Cosine
3. Lemmas with Overlap

The results of the SVM-gen baseline for the RTE-3 development dataset with a 10-fold-cross-validation is 63.8% accuracy. The results for SVM-gen with and without feature selection are similar in the 10-fold-cross-validation.

The experimental results for both SVM-gen versions are summarised in Table 3.2. In addition, we compute the McNemar's test on both SVM-gen versions, and there is no statistically significant difference between them.

Table 3.2: Accuracy results for the SVM-gen baseline system with all the features and with the selected features

Method	RTE-1	RTE-2	RTE-3
SVM-gen	51.38%	58.5%	59.5%
SVM-gen with feature selection	50.75%	58.37%	59%
Official baseline	49%	53%	57%

The SVM-gen baseline with feature selection shows worst results on the test datasets. A possible reason for this results is that the genetic algorithm found an over-fitted solution from the development dataset.

Table 3.3: The 10-fold cross-validation accuracy results over the RTE development datasets for the ML-TINE model

Algorithm	RTE-1	RTE-2	RTE-3
ML-TINE	**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%

We use the RTE-1, RTE-2, and RTE-3 development datasets to train different classifiers for the ML-TINE model. Table 3.3 shows the 10-fold cross-validation results for different ML algorithms using the same feature set of ML-TINE. The SVM algorithm achieved the best results in the experiments during the training phase. We use this algorithm to perform the classification over the RTE test datasets (i.e. ML-TINE).

The data used for classification are the test datasets of the RTE Challenge. The experimental results and comparison with previous work are summarised in Table 3.4.

Table 3.4: Accuracy comparison against previous work over the RTE test datasets for the ML-TINE model

Method	RTE-1	RTE-2	RTE-3
Roth and Sammons (2007)	-	-	65.56%
Burchardt and Frank (2006), Burchardt et al. (2007)	54.6%	59.8%	62.62%
Delmonte et al. (2005), Delmonte et al. (2006), Delmonte et al. (2007)	59.25%	54.75%	58.75%
ML-TINE	53.87%	55.37%	61.75%
SVM-gen	50.75%	58.37%	59%
Official baseline	49%	53%	57%

However, previous work are more complex systems. In contrast, ML-TINE relies in scoring predicate-argument information with the support of simple string metrics. Our main semantic feature is focused in predicate-argument information, where other methods tackle several semantic phenomena such as negation and discourse information (Roth and Sammons, 2007), or exploit a large number of other types of features (Burchardt et al., 2007).

Table 3.5: Comparison with overall accuracy results over the RTE test datasets for the ML-TINE model

Challenge	ML-TINE	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%

Table 3.5 shows the overall accuracy results of the RTE test datasets against our method. Our method ML-TINE is close to the average performance but far from the best method.

For the ML-EDIT model we use the same configuration for the SVM as the SVM-gen baseline. The experimental results for the ML-EDIT model using the edit distance metric with the chunking metric backoff are summarised in Table 3.6. The backoff score is used if the edit score between a T–H pair is zero. We also train a second ML-EDIT model using a a feature set consisting of simple lexical extra features and the edit distance metric, where the simple features are the same as those used in ML-TINE.

Table 3.6: Accuracy results for ML-EDIT model over the test datasets

Method	RTE-1	RTE-2	RTE-3
ML-EDIT plus backoff	50.25%	51.87%	51.25%
ML-EDIT plus lexical features	51.5%	57.87%	59.37%

ML-EDIT based on lexical features outperforms the backoff strategy. We use the method based on lexical features for further comparison.

Table 3.7: Comparison of ML-EDIT with overall accuracy results over the RTE test datasets

Challenge	ML-EDIT	Average	Best
RTE-1	51.5%	55.12%	70.00%
RTE-2	57.87%	58.62%	75.38%
RTE-3	59.37%	61.14%	80.00%

Table 3.7 shows the overall accuracy results of the RTE test datasets against our method. ML-EDIT is close to the average performance but far from the best method. However, our method ML-EDIT shows very competitive results on the RTE 2 and 3 datasets.

Table 3.8: Accuracy comparison with previous works over the RTE test datasets for the ML-EDIT

Method	RTE-1	RTE-2	RTE-3
Roth and Sammons (2007)	-	-	65.56%
Burchardt and Frank (2006), Burchardt et al. (2007)	54.6%	59.8%	62.62%
Delmonte et al. (2005), Delmonte et al. (2006), Delmonte et al. (2007)	59.25%	54.75%	58.75%
Kouylekov and Magnini (2005), Kouylekov and Magnini (2006)	55%	60%	-
ML-EDIT	51.5%	57.87%	59.37%
SVM-gen	50.75%	58.37%	59%
Official baseline	49%	53%	57%

Table 3.8 shows the comparison with previous work. ML-EDIT is comparable with Kouylekov and Magnini (2005). However, ML-EDIT improves in a small degree the proposed baseline with feature selection (i.e. SMV-gen). Thus, ML-EDIT reduces the amount of errors with additional semantic-scored pairs (aligned and edited pairs). In addition, the remaining pairs out of the coverage of the semantic score are scored by the back-off, and this back-off shows to be a poor predictor for entailment.

In the remainder of this section, we show the comparison of the Statistical Relational Learning models with previous work. Table 3.9 shows the performance of our MLN-BASELINE and MLN-RELATIONAL against that

of the official baseline. It also shows the top system and the average accuracy scores for all systems reported in the RTE challenges.

Table 3.9: Comparison of MLN-BASELINE and MLN-RELATIONAL with overall accuracy results over the RTE test datasets

Method	RTE-1	RTE-2	RTE-3
Top system	70%	75%	80%
Avg. systems	55%	59%	61%
Official baseline	49%	53%	57%
MLN-BASELINE	56%	54%	51%
MLN-RELATIONAL	57%	55%	65%

The MLN-RELATIONAL achieves a competitive performance compared to the average of the participating systems, particularly on the RTE-1 dataset (Avg. systems). However, its performance is far from that of the best system (Top).

Table 3.10: Comparison of MLN-BASELINE and MLN-RELATIONAL with the Propositional learning models

Method	RTE-1	RTE-2	RTE-3
ML-TINE	53.87%	55.37%	61.75%
ML-EDIT	51.5%	57.87%	59.37%
SVM-gen	50.75%	**58.37%**	59%
Official baseline	49%	53%	57%
MLN-BASELINE	56%	54%	51%
MLN-RELATIONAL	**57%**	55%	**65%**

Table 3.10 shows the comparison of the MLN models with the propositional models. The MLN models outperform the propositional models on the RTE 1 and 3 but on the RTE 1 the simple SVM-gen baseline outperforms all the proposed models. The RTE-2 dataset shows to be a hard evaluation for

methods using semantic information. For example, the method of Delmonte et al. (2006) achieves an accuracy of 59.25%, Burchardt and Frank (2006) an accuracy of 59.8% and a strong alignment-based method (de Marneffe et al., 2006) achieves an accuracy of 60.5% on the same dataset. The average accuracy of methods for the RTE-2 dataset is 59%.

For a fair comparison, we evaluate our MLN-RELATIONAL method against all previous work for RTE that is also based on alignment techniques. de Marneffe et al. (2006) use a two-stage alignment similar to ours, but using 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. Then, they perform training and testing with a logistic regression classifier. Chambers et al. (2007) improve the alignment stage in (de Marneffe et al., 2006) and combine it with a logical framework for the second stage (MacCartney and Manning, 2007). 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. (2008) propose a phrase-base alignment that uses external lexical resources. They improve the first stage via knowledge about semantic similarity and a specific dataset only to train the alignment.

Table 3.11: Accuracy against previous work based on alignment over the RTE datasets for the MLN-RELATIONAL model

Method	RTE-1	RTE-2	RTE-3
de Marneffe et al. (2006)	-	60.5%	60.5%
Chambers et al. (2007)	-	-	63.62%
MacCartney et al. (2008)	-	60.3%	-
MLN-RELATIONAL	57%	55%	65%

Table 3.11 shows that our MLN-RELATIONAL method 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 low performance of our alignment method. TINE Context Matching 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 method 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, for which our method can find a good quality alignment because of the larger context. This also seem to affect the overall performance of the participating systems, since the average accuracy for RTE-1 is 55%, 59% for RTE-2, and 61% for RTE-3.

Our method 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 can be because TINE Context Matching finds more alignments for the *TRUE* pairs.

Furthermore, we test how our model behave over a subset of the datasets for which TINE Context Matching produces an alignment. We compare the relational model only with the alignment features (i.e. without the shallow features) against the official baseline. We compare our MLN methods with a widely used SVM baseline (Mehdad and Magnini, 2009) as well as of our in-house baseline SVM-gen. Table 3.12 shows the results, where the relational model clearly outperforms the official baseline, and by a large margin on the RTE-3 dataset. This shows the potential of the relational features and MLNs for RTE.

Table 3.12: 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
Official baseline	50%	51%	56%
SVM-gen	50.75%	58.37%	59%
MLN-RELATIONAL	57%	55%	78%

The results over this subset are similar on the RTE 1 and 2 datasets compared to the full model over the complete RTE 1 and 2 datasets. Therefore, the shallow features decrease the performance in the RTE 3 dataset with the inclusion of instances without a semantic structure.

For a comparison covering the other main aspect of our method – its probabilistic nature, in a second evaluation experiment, in Table 3.13 we compare our method against other methods based on probabilistic modelling.

Glickman and Dagan (2006) model entailment via lexical alignment, where the web co-occurrences for a pair of words are used to describe the probabil-

ity of the hypothesis given the text. Harmeling (2007) propose a model that, with a given sequence of transformations over a parse tree, keeps entailment decisions with a certain probability. Wang and Manning (2010) 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. (2013) extend the work in (Garrette et al., 2011) to be able to process large scale datasets such as those from the RTE challenges. The method transforms distributional similarity judgements into weighted inference formulas, where the distributional similarity describes a degree of entailment between pairs (i.e., If X and Y occur in similar contexts they describe similar entities).

Table 3.13: Accuracy against previous work based on probabilistic modelling over the RTE datasets for the MLN-RELATIONAL model

Method	RTE-1	RTE-2	RTE-3
Glickman and Dagan (2006)	59%	-	-
Harmeling (2007)	-	-	59.3%
Wang and Manning (2010)	-	63%	61.1%
Beltagy et al. (2013)	57%	-	-
MLN-RELATIONAL	57%	55%	65%

Table 3.13 shows a similar behaviour as the previous comparison: considerably better results on RTE-3, but lower results for RTE-2. In addition, for the RTE-1 dataset, which has also been used by most of these other methods, our relational model shows very competitive performance. In particular, our method achieves the same performance as Beltagy et al. (2013), which also uses an MLN for the entailment decision, as mentioned in Chapter 2.

3.5 Summary

The TINE Lexical Matching and TINE Context Matching alignment methods give as output an alignment matrix, as well as similarity score between a semantic representation of the T-H pair. The TINE Edit Distance gives as output only a similarity score. We use these sources of information to train two different entailment models: i) a propositional learning model and ii) a statistical learning model. The methods show promising results compared to previous work. However, the coverage of the alignment methods affects the overall performance.

Chapter 4

Methods for Measuring the Bidirectional Equivalence of Texts

In this Chapter, we use our proposed alignment methods in two NLP applications, namely MT evaluation and STS. These applications serve as an extrinsic evaluation for measuring the impact of our alignment methods. In MT evaluation, the alignment is used as part of a metric to assess machine translations against human translations. In STS, the alignment plays the role of a feature to score the similarity between text pairs. In this section we also describe our proposed method for STS based on MTL.

4.1 Alignment-based Machine Translation Evaluation

The automatic evaluation of MT is a long-standing problem. A number of metrics have been proposed in the last two decades, mostly measuring some form of matching between the machine translation (hypothesis) and one or more human (reference) translations. However, most of these metrics focus on fluency aspects, as opposed to adequacy (meaning). Therefore, measuring whether the meaning of the hypothesis and reference translation are the same or similar is still an understudied problem.

The most commonly used metrics, BLEU (Papineni et al., 2002) and alike, perform simple exact matching of n-grams between hypothesis and reference translations. Such a simple matching procedure has well known limitations, including that the matching of non-content words counts as much as the matching of content words, that variations of words with the same meaning are disregarded, and that a perfect matching can happen even if the order of sequences of n-grams in the hypothesis and reference translation are very different, changing completely the meaning of the translation.

A few metrics have been proposed in recent years to address the problem of measuring whether a hypothesis and a reference translation share the same meaning. The most well-know metric is probably METEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2010). METEOR is based on a generalised concept of unigram matching between the hypothesis and the reference translation. Alignments are based on exact, stem, synonym, and paraphrase matches between words and phrases. However, the structure of the sentences is not considered, but similar word orders are rewarded through higher scores for the matching of longer fragments.

Wong and Kit (2010) measure word choice and word order by the matching of words based on surface forms, stems, senses and semantic similarity. The informativeness of matched and unmatched words is also weighted.

Liu et al. (2010) propose to match bags of unigrams, bigrams and trigrams considering both recall and precision and F-measure giving more importance to recall, but also using WordNet synonyms.

Tratz and Hovy (2008) use transformations in order to match short syntactic units defined as Basic Elements (BE). The BE are minimal-length syntactically well defined units. For example, nouns, verbs, adjectives and adverbs can be considered BE-Unigrams, while a BE-Bigram could be formed from a syntactic relation (e.g. subject+verb, verb+object). BEs can be lexically different, but semantically similar.

Padó et al. (2009a) use Textual Entailment features extracted from the Stanford Entailment Recogniser (MacCartney et al., 2006). The Textual Entailment Recogniser computes matching and mismatching features over dependency parses. The metric predicts the MT quality with a regression model. The alignment is improved using ontologies.

He et al. (2010) measure the similarity between hypothesis and reference translation in terms of the Lexical Functional Grammar (LFG) representation. The representation uses dependency graphs to generate unordered sets of dependency triples. Calculating precision, recall, and F-score on the sets of triples corresponding to the hypothesis and reference segments allows measuring similarity at the lexical and syntactic levels. The measure also matches WordNet synonyms.

Castillo and Estrella (2012) propose and approach based on a STS setup. The metric is based on similarity metrics extracted from WordNet. The metric produces feature vectors for the language pairs ES-EN, DE-EN, FR-EN and CS-EN. The method takes advantage of available resources for EN (i.e. WordNet).

Wang and Manning (2012) propose a metric that computes probabilistic edit distance used as predictions to evaluate MT translation outputs. The method learns the weights of edit distance operations for a finite state model. The major contribution of this method is the extension of edit operations based on phrase shift and word swapping. The finite state model accept the sequence of edit operations that transforms the reference translation into the system translation.

Wu et al. (2013b) propose the use of syntax features in the source side to evaluate MT. The metric computes distance between paths from dependency trees. This metric outperforms the most common metrics such as BLEU and METEOR.

For a more detailed description of metrics we refer the reader to (Callison-Burch et al., 2012; Macháček and Bojar, 2013; Machacek and Bojar, 2014).

The closest related metric to TINE Lexical Matching is that by Lo and Wu (2011), which is a semi automated metric based on the matching of semantic role fillers. First, the translations are annotated with SRL manually or automatically. The semantic frames between the reference and translation are compared frame by frame and argument by argument. The frame score is the weighted sum of the correctly translated arguments. The metric score is defined by the f-score where the precision and recall is computed by the average of the translation accuracy for all the frames in the system translation over the number of frames in the reference translation.

Giménez et al. (2010) also uses shallow semantic representations. Such a metric combines a number of components, including lexical matching metrics like BLEU and METEOR, as well as components that compute the matching of constituent and dependency parses, named entities, discourse representations and semantic roles. However, the semantic role matching is based on exact matching of roles and role fillers. Moreover, it is not clear what the contribution of this specific information is for the overall performance of the metric.

4.1.1 Metric Description

We use TINE Lexical Matching as an adequacy component in order to deal with both word choice and semantic structure. Additionally, TINE Lexical Matching uses an exact lexical matching component to reward hypotheses that present the same lexical choices as the reference translation. The overall score s is defined using the simple weighted average model in Equation (4.1):

$$s(H, \mathbf{R}) = max \left\{ \frac{\alpha L(H, R) + \beta A(H, R)}{\alpha + \beta} \right\}_{R \in \mathbf{R}} \tag{4.1}$$

where H represents the hypothesis translation, R represents a reference translation contained in the set of available references $\mathbf{R}$; L defines the (exact) lexical match component in Equation (4.2), A defines the adequacy component in Eq. (3.1); and α and β are tunable weights for these two components. In Eq. (3.1) we use as variables: H hypothesis translation and R reference translation instead of hypothesis H and text T from RTE. If multiple refer-

ences are provided, the score of the segment is the maximum score achieved by comparing the segment to each available reference.

$$L(H,R) = \frac{|H \bigcap R|}{\sqrt{|H| * |R|}} \tag{4.2}$$

The lexical match component measures the overlap between the two representations in terms of the cosine similarity metric. A segment, either a hypothesis or a reference, is represented as a bag of tokens extracted from an unstructured representation, that is, bag of unigrams (words or stems).

Cosine similarity was chosen, as opposed to simply checking the percentage of overlapping words (POW), because cosine similarity does not penalise differences in the length of the hypothesis and reference translation as much as POW. Cosine similarity normalises the cardinality of the intersection $|H \cap R|$ using the geometric mean $\sqrt{|H| * |R|}$ instead of the union $|H \cup R|$. This is particularly important for the matching of arguments - which is also based on cosine similarity. If an hypothesised argument has the same meaning as its reference translation, but differs from it in length, cosine will penalise less the matching than POW. That is specially interesting when core arguments get merged with modifiers due to bad semantic role labelling (e.g. *[A0 I] [T bought] [A1 something to eat yesterday]* instead of *[A0 I] [T bought] [A1 something to eat] [AM-TMP yesterday]*).

4.1.2 Machine Translation Metric Evaluation Datasets

The evaluation task estimates the performance of automatic evaluation metrics for machine translation (Callison-Burch et al., 2012). The organisers provide translations produced by the participating systems in the translation task along with the reference human translations. The automatic metrics rank each of the translations at the system-level or at the segment-level. The metrics performance is evaluated using the correlation between the predicted scores (i.e. rankings) with the human judgements.

The evaluation datasets consist of the output of machine translation systems for five different language pairs (e.g. French-English, Spanish-English, German-English, Czech-English) along with the reference translations for each language pair and the respective ranking annotation. The metric computes scores for each of the outputs at the system-level and the segment-level. The correlation is measured as follows:

System-level correlation Spearman's rank correlation coefficient (rho) to measure the correlation of the automatic metrics with the human judgements of translation quality at the system-level. The system rank will be assigned based on the percent of time that the sentences it produced were judged to be better than or equal to the translations of any other system. The automatic metrics scores are converted into rankings before calculating the correlation.

Segment-level correlation Kendall's tau to measure metrics' correlation with human judgements at the sentence-level. For every pairwise comparison of the output of two systems for a sentence the correlation is computed by counting if the metric score (i.e. decision) is concordant with the human judgement if the metric orders the output in the same way. In other words, if the metric gives a higher score to a higher ranked system. The ties are not counted for correlation.

The datasets are created by translators using news articles, where the annotator agreement is measured with the kappa coefficient. The inter-annotator agreement ranges from 0.176 kappa to 0.336, while intra-annotator agreement ranges from 0.279 to 0.648 kappa. The WMT08, WMT09 and WMT10 datasets (Callison-Burch et al., 2012) with manually annotated rankings are used by the automatic metrics to tune their parameters. The baseline metric is BLEU, which achieves an average correlation of 0.53 for system-level and 0.17 for segment-level.

4.1.3 Results and Discussion

We set the weights α and β by experimental testing to $\alpha = 1$ and $\beta = 0.25$. The lexical component weight is prioritised because it has shown a good average Kendall's tau correlation (0.23) on a development dataset (Callison-Burch et al., 2010). Table 4.1 shows the correlation of the lexical component with human judgements for a number of language pairs.

We use the SENNA SRL system to tag the dataset with semantic roles.

Table 4.1: Kendall's tau segment-level correlation of the lexical component with human judgements

Metric	cz-en	fr-en	de-en	es-en	avg
$L(H, R)$	0.27	0.21	0.26	0.19	0.23

We discuss with a few examples some of the common errors made by the TINE Lexical Matching method for MT evaluation. The errors made by our alignment methods are also present on the previous RTE results. Overall, we consider the following categories of errors:

1. Lack of coverage of the ontologies.

   H: *This year, women were awarded the Nobel Prize in all fields except physics.*

   R: *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.

   H: *If snow falls on the slopes this week, Christmas will sell out too, says Schiefert.*

R: *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 (Chklovski and Pantel, 2004) was created by a semi-automatic extraction algorithm with an average accuracy of 65.5%.

3. Incorrect tagging of the semantic roles by the semantic parser SENNA.

H: *Colder weather is forecast for Thursday, so if anything falls, it should be snow.*
R: *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*.

We compare TINE Lexical Matching against standard BLEU, METEOR (Denkowski and Lavie, 2010) and other previous metrics reported in (Callison-Burch et al., 2010) which also claim to use some form of semantic informa-

tion. The comparison is made in terms of Kendall's tau correlation against the human judgements at a segment-level. For our submission to the shared evaluation task (i.e. (Rios et al., 2011)), system-level scores are obtained by averaging the segment-level scores. Our metric achieved rank *13* at segment-level with an average correlation of *0.23* and rank *2* at system-level with an average correlation of *0.87*. Our metric outperforms most the previous work at system-level. A full comparison of metrics at system/segment level is reported in (Callison-Burch et al., 2011).

Table 4.2: Comparison with common metrics and previous semantically-oriented metrics using segment-level Kendall's tau correlation with human judgements

Metric	cz-en	fr-en	de-en	es-en	avg
Liu et al. (2010)	0.34	0.34	0.38	0.34	0.35
Giménez et al. (2010)	0.34	0.33	0.34	0.33	0.33
Wong and Kit (2010)	0.33	0.27	0.37	0.32	0.32
METEOR	0.33	0.27	0.36	0.33	0.32
TINE Lexical Matching	0.28	0.25	0.30	0.22	0.26
BLEU	0.26	0.22	0.27	0.28	0.26
He et al. (2010)	0.15	0.14	0.17	0.21	0.17
Tratz and Hovy (2008)	0.05	0.0	0.12	0.05	0.05

TINE Lexical Matching achieves the same average correlation with BLEU, but outperforms it for some language pairs. Additionally, TINE outperforms

some of the other metrics which use WordNet to deal with synonyms as part of the lexical matching.

The closest metric to TINE Lexical Matching on this dataset is (Giménez et al., 2010), which also uses semantic roles as one of its components, achieves better performance. However, this metric is a rather complex combination of a number of other metrics to deal with different linguistic phenomena. The metric uses lexical, syntactical and semantic information, which the latest version has more than 600 metrics (i.e. different variants of each metric). The lexical representation the uses variants of BLEU, METEOR, ROUGE, and TERp. The syntactic representation metrics are: i) shallow parsing metric that is the average lexical overlap over parts of speech and base phrase chunk types, ii) dependency parsing metric that is the head matching over word forms, grammatical categories and relations, and average lexical overlap between tree nodes according to their tree level, category or relation, and iii) constituency parsing metric that is average lexical overlap over parts of speech and syntactic constituents, and syntactic tree matching. The semantic representation metrics are: i) named entities metric that is the average lexical overlap between NERs according to their type, ii) semantic roles metric that is the average lexical overlap between frames according to their type, and average role overlap, and iii) discourse representation metric that is the average lexical overlap over discourse representations according to their type.

As an additional experiment, we use BLEU as the lexical component $L(H, R)$ in order to test if the shallow-semantic component can contribute

to the performance of this standard evaluation metric. Table 4.3 shows the results of the combination of BLEU and TINE Lexical Matching defined as TINE-B by using the same parameter configuration as above. The addition of TINE Lexical Matching increased the average correlation of BLEU from 0.26 to 0.28.

Table 4.3: TINE-B: Combination of BLEU and the shallow-semantic component

Metric	cz-en	fr-en	de-en	es-en	avg
TINE-B	0.27	0.25	0.30	0.30	0.28

The best contribution of the TINE-B combination is on the es-en language pair, where the solely use of TINE achieves a correlation of 0.22 and BLEU 0.28. Thus, TINE is helping BLEU to improve the overall performance in a similar way as the work of (Giménez et al., 2010), where different types of metrics cope with several linguistic phenomena. Finally, we attempt to improve the tuning of the weights of the components (α and β parameters) by using a simple genetic algorithm (Back et al., 1999) to select the weights that maximise the correlation with human scores on a development set (we use the development sets from WMT10 (Callison-Burch et al., 2010)).

The configuration of the genetic algorithm is as follows:

- Fitness function: Kendall's tau correlation
- Chromosome: two real numbers, α and β
- Number of individuals: 80

- Number of generations: 100
- Selection method: roulette
- Crossover probability: 0.9
- Mutation probability: 0.01

We use a similar configuration for the genetic algorithm as in Chapter 3. We decrease the number of generations given the large amount of sentences present in the WMT datasets (i.e. 2034 sentences for each language pair). However, we increase the probability of crossing to give a chance to a larger amount of solutions (i.e. individuals) to contribute during search.

Table 4.4: Optimised values of the parameters using a genetic algorithm and Kendall's tau correlation of the metric on the test sets

Language pair	Correlation	α	β
cz-en	0.28	0.62	0.02
fr-en	0.25	0.91	0.03
de-en	0.30	0.72	0.1
es-en	0.31	0.57	0.02
avg	0.29	–	–

Table 4.4 shows the parameter values obtaining from tuning for each language pair and the correlation achieved by the metric with such parameters. With such an optimisation step the average correlation of the metric increases to 0.29. The addition of the shallow-semantic component into a lexical component yields absolute improvements in the correlation of 3%-6% on average, depending on the lexical component used (cosine similarity or BLEU).

4.2 Alignment-based Semantic Textual Similarity

For STS, we train a regression algorithm with different types of similarity metrics as features: i) lexical, ii) syntactic and iii) semantic. The lexical similarity metrics are: i) cosine similarity using a BoW representation, and ii) precision, recall and F-measure of content words.

The syntactic metric computes BLEU over the labels of base-phrases (chunks) instead of words, two semantic metrics are used: a metric based on the preservation of named entities and TINE Context Matching. Named entities are matched by type and content: while the type has to match exactly, the content is compared with the assistance of a distributional thesaurus. Finally, we use METEOR, that computes inexact word overlap.

The lexical and syntactic metrics complement the semantic metrics in dealing with the phenomena observed in the task's dataset. For example, from the MSRvid dataset:

S1 Two men are $[\text{playing}]_V$ football.

S2 Two men are $[\text{practicing}]_V$ football.

In this case, as typical of paraphrasing, the situation and participants are the same while the surface realisation differs, but *playing* can be considered similar to *practicing*. From the SMTeuroparl dataset:

S3 The Council of Europe, along with the [Court of Human Rights]$_{AM-ADV}$, [has]$_{V}$ a wealth of experience of such forms of supervision, and we can build on these.

S4 Just as the [European Court of Human Rights]$_{AM-ADV}$, the Council of Europe [has]$_{V}$ also considerable experience with regard to these forms of control; we can take as a basis.

Similarly, here although with different realisations, the *Court of Human Rights* and the *European Court of Human Rights* represent the same entity.

Semantic metrics based on predicate-argument structure can play a role in cases when different realisation have similar semantic roles:

S5 The right of a government arbitrarily to set aside its own constitution is the defining characteristic of a tyranny.

S6 The right for a government to draw aside its constitution arbitrarily is the definition characteristic of a tyranny.

For sentence S5 the SRL annotation is as follows:

right The [right]$_{V}$ [of a government]$_{A0}$ [arbitrarily to set aside its own constitution]$_{A1}$ is the defining characteristic of a tyranny.

set The right of [a government]$_{A0}$ [arbitrarily]$_{AM-MNR}$ to [set]$_{V}$ [aside]$_{A2}$ [its own constitution]$_{A1}$ is the defining characteristic of a tyranny.

define The right of a government arbitrarily to set aside its own constitution is [the]$_{A0}$ [defining]$_{V}$ [characteristic of a tyranny]$_{C-A0}$.

characteristic The right of a government arbitrarily to set aside its own constitution is the [defining]$_{A1}$ [characteristic]$_{V}$ [of a tyranny]$_{A1}$.

For sentence S6 the SRL annotation is as follows:

right The [right]$_{V}$ [for a government to draw aside its constitution]$_{A0}$ arbitrarily is the definition characteristic of a tyranny.

draw The right for [a government]$_{A0}$ to [draw]$_{V}$ [aside]$_{AM-DIR}$ [its constitution]$_{A1}$ arbitrarily is the definition characteristic of a tyranny.

characteristic The right for a government to draw aside its constitution arbitrarily is the [definition]$_{A1}$ [characteristic]$_{V}$ [of a tyranny]$_{A1}$.

4.2.1 Features Description

All our lexical metrics use the same surface representation: words. However, the cosine metric uses BoW, while all the other metrics use only content words. We thus first represent the sentences as BoW. For example, given the pair of sentences S7 and S8:

S7 A man is riding a bicycle.

S8 A man is riding a bike.

the BoW's are S7 = {A, man, is, riding, a, bicycle,.} and S8 = {A, man, is, riding, a, bike, .}, and the bag-of-content-words are S7 = {man, riding, bicycle} and S8 = {man, riding, bike}.

We compute similarity scores using the following metrics between a pair of sentences A and B: cosine distance (Eq. 3.6), precision (Eq. 3.7), recall (Eq. 3.8) and F-measure (Eq. 3.9).

The BLEU metric computes the precision of exact matching of n-grams between a hypothesis and reference translations. This simple procedure has limitations such as: the matching of non-content words mixed with the counts of content words affects in a perfect matching that can happen even if the order of sequences of n-grams in the hypothesis and reference translation are very different, changing completely the meaning of the translation. To account for similarity in word order we use BLEU over base-phrase labels instead of words, leaving the lexical matching for other lexical and semantic metrics. We compute the matchings of 1-4-grams of base-phrase labels. This metric favours similar syntactic order.

The METEOR metric, previously described, is also a lexical metric based on unigram matching between two sentences. However, matches can be exact, using stems, synonyms, or paraphrases of unigrams.

TINE Context Matching is an automatic metric based on the use semantic roles to align predicates and their respective arguments in a pair of sentences.

We use the following state-of-the-art tools to pre-process the data for feature extraction: i) TreeTagger[1] for lemmas and ii) SENNA for Part-of-Speech tagging, Chunking, Named Entity Recognition and Semantic Role Labelling.

[1]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

The complete feature set includes:

- Lexical metrics
  - Cosine metric over BoW
  - Precision over content words
  - Recall over content words
  - F-measure over content words
- BLEU metric over chunks
- METEOR metric over words (with stems, synonyms and paraphrases)
- Named entity metric Eq. (3.11)
- Semantic Role Labelling metric (i.e. TINE Context Matching Eq. (3.3))

The Machine Learning algorithm used for regression is the LIBSVM[2] Support Vector Machine (SVM) implementation using the radial basis kernel function (RBF). We used a simple genetic algorithm to tune the parameters of the SVM. We use a similar configuration for the genetic algorithm as in Chapter 3, but we modify the parameters of the genetic algorithm to fit this task. For example, less generations for faster results. The configuration of the genetic algorithm is as follows:

[2]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

- Fitness function: minimise the mean squared error found by cross-validation
- Chromosome: real numbers for SVM parameters γ, *cost* and ϵ
- Number of individuals: 80
- Number of generations: 100
- Selection method: roulette
- Crossover probability: 0.9
- Mutation probability: 0.01

4.2.2 Semantic Textual Similarity Evaluation Datasets

The participating methods in the STS evaluation challenge have to compute a similarity score between the pair of sentences S1 and S2. The participants are allowed to send a limited number of different versions of their main proposed method, which are called *runs*. The similarity scores range between 0 and 5 for each pair of sentences, given the following official guidelines:

- Score (5) The two sentences are completely equivalent, as they mean the same thing. For example,

  The bird is bathing in the sink.

  Birdie is washing itself in the water basin.

- Score (4) The two sentences are mostly equivalent, but some unimportant details differ. For example,

In May 2010, the troops attempted to invade Kabul.

The US army invaded Kabul on May 7th last year, 2010.

- Score (3) The two sentences are roughly equivalent, but some important information differs/missing. For example,

  John said he is considered a witness but not a suspect.

  "He is not a suspect anymore." John said.

- Score (2) The two sentences are not equivalent, but share some details. For example,

  They flew out of the nest in groups.

  They flew into the nest together.

- Score (1) The two sentences are not equivalent, but are on the same topic. For example,

  The woman is playing the violin.

  The young lady enjoys listening to the guitar.

- Score (0) The two sentences are on different topics.

  John went horse back riding at dawn with a whole group of friends.

  Sunrise at dawn is a magnificent view to take in if you wake up early enough for it.

The similarity scores are rated by human judges using Amazon Mechanical Turk based on the 0-5 score range. The results of the methods performance is evaluated using the *Pearson product-moment correlation coefficient*

between the method scores and the human scores. The agreement of each annotator with the average scores was between 87% and 89%. Similar to RTE the STS task also provides a unified framework that accounts for the impact of an extrinsic evaluation of multiple semantic components.

The first STS challenge in 2012 (Agirre et al., 2012) used training and test datasets that contain sentence pairs from publicly available paraphrase and machine translation resources. The MSR-Paraphrase, Microsoft Research Paraphrase Corpus (MSRPar) were used to evaluate text similarity algorithms. The MSR-Video, Microsoft Research Video Description Corpus (MSRVid), where the authors showed brief video segments to annotators from Amazon Mechanical Turk and they were asked to provide a one-sentence description of the main action or event in the video. Each of the previous datasets include 1500 sentence pairs each. As well as for machine translation evaluation 1500 sentence pairs. The mapping between OntoNotes and WordNet senses (OnWN) have 750 sentence pairs.

A common approach for STS is based on using similarity metrics as features to train regression algorithms. For example, the baseline approach consists in using a simple word overlap. The input sentences are preprocessed with tokenisation and splitting at white spaces, and each sentence is represented as a vector in a multidimensional token space. Each dimension have value 1 if the token is present in the sentence, 0 otherwise. The similarity score of the previous vectors is computed by using cosine similarity. On the STS 2012 the correlation results of the official baseline were as follows:

OnWN r=0.5864, MSRpar r=0.4334 and MSRvid r=0.2996, and the results on the MT evaluation task are for SMT-europarl r=0.4542 and SMT-news r=0.3908.

For the STS 2013 (Agirre et al., 2013) the participating methods had to cope with the additional challenge of the lack of training data, namely CORE task. The CORE task is similar to the STS 2012 task, but with datasets from different tasks compare to 2012. The included tasks are: news wire headlines, machine translation evaluation datasets and multiple lexical resource glossed sets. The headlines dataset is composed by naturally occurring news headlines gathered by the Europe Media Monitor engine[3] from several different news sources. The OnWN/FnWN lexical recourse dataset contains gloss pairs from two sources: OntoNotes-WordNet (OnWN) and FrameNet-WordNet (FnWN). The baseline is the same as in 2012, but with the addition of the best 2012 systems (Šarić et al., 2012; Bär et al., 2012).

Also, the new typed-similarity (TYPED) task was proposed. The TYPED task is based on computing the similarity between items, using textual metadata. The participants need to score specific types of similarity, like similar author, similar time period among others. The TYPED dataset is annotated by using crowdsourcing with 1500 pairs, where the dataset is divided in 750 pairs for training and 750 for test.

[3]http://emm.newsbrief.eu/overview.html

In the STS 2014 (Agirre et al., 2014) new tasks were proposed to evaluate the participating systems[4]. The new tasks are: deft discussion forum and news (deft-forum and deft-news), deft-news (news article data in the DEFT project), image description (image), news title and tweet comments (tweet-news). A new subtask was proposed, namely Spanish STS. The goal of this subtask is evaluating STS systems for Spanish. As well as in the 2013 STS both subtask evaluation challenges lack of a training dataset.

4.2.3 Results and Discussion

We have three different system runs (run1, run2, run3), each is a variation of the above feature set. For the official submission to STS 2012 we used the systems with optimised SVM parameters. We trained SVM models with each of the following STS 2012 task datasets: MSRpar, MSRvid, SMTeuroparl and the combination of MSRpar+MSRvid. For each test dataset we applied their respective training task, except for the unseen test tasks, not covered by any training task: for On-WN we used the combination MSRpar+MSRvid, and for SMTnews we used SMTeuroparl.

Tables 4.5 to 4.7 show the Pearson correlation of our three systems/runs for individual datasets of the predicted scores against human annotation, compared against the official baseline.

Our first run (run1) uses the lexical, BLEU, METEOR and named entities features, without the TINE Context Matching feature. Table 4.5 shows the

[4]http://alt.qcri.org/semeval2014/task10/

results over the test set, where run1-A is the version without SVM parameter optimisation and run1-B are the official results with optimised parameters for SVM.

Table 4.5: Results STS 2012 for run1 using lexical, chunking, named entities and METEOR as features. A is the non-optimised version, B are the official results

Task	run1-A	run1-B	Baseline
MSRpar	0.455	0.455	0.433
MSRvid	0.706	0.362	0.300
SMTeuroparl	0.461	0.307	0.454
OnWN	0.514	0.281	0.586
SMTnews	0.386	0.208	0.390

In run1 we use only the TINE Context Matching feature in order to analyse whether this feature on its own could be sufficient or lexical and other simpler features are important. Table 4.6 shows the results over the test set without parameter optimisation (run2-A) and the official results with optimised parameters for SVM (run2-B).

Table 4.6: Results STS 2012 for run2 using the SRL feature only. A is the non-optimised version, B are the official results

Task	run2-A	run2-B	Baseline
MSRpar	0.335	0.300	0.433
MSRvid	0.264	0.291	0.300
SMTeuroparl	0.264	0.161	0.454
OnWN	0.281	0.257	0.586
SMTnews	0.189	0.221	0.390
ALL	0.096	0.536	0.311

In run3 we use all features. Table 4.7 shows the results over the test set without parameter optimisation (run3-A) and the official results with optimised parameters for SVM (run3-B).

Table 4.7: Results for run3 using all features. A is the non-optimised version, B are the official STS 2012 results

Task	run3-A	run3-B	Baseline
MSRpar	0.472	0.353	0.433
MSRvid	0.705	0.572	0.300
SMTeuroparl	0.471	0.307	0.454
OnWN	0.511	0.264	0.586
SMTnews	0.410	0.116	0.390

Table 4.8 shows the average results over all five datasets, where ALL stands for the Pearson correlation with the gold standard for the five dataset, Rank is the absolute rank among all submissions, ALLnrm is the Pearson correlation when each dataset is fitted to the gold standard using least squares, RankNrm is the corresponding rank and Mean is the weighted mean across the five datasets, where the weight depends on the number of sentence pairs in the dataset.

Table 4.8: Official STS 2012 results and ranking over the test set for runs 1-3 with SVM parameters optimised

System	ALL	Rank	ALLnrm	RankNrm	Mean	RankMean
run1	0.640	36	0.719	71	0.382	80
run2	0.536	59	0.629	88	0.257	88
run3	0.598	49	0.696	82	0.347	84
Baseline	0.311	87	0.673	85	0.436	70

Table 4.8 shows the ranking and normalised official scores of our official submissions compared against the baseline. Our submissions outperform the official baseline but significantly underperform the top systems in the shared task. The best system (run1) achieved an above average ranking, but disappointingly the performance of our most complete system (run3) using the semantic metric is poorer. Surprisingly, the non-optimised versions outperform the optimised versions used in our official submission. One possible reason for that is the overfitting of the optimised models to the training sets.

Our run1 and run3 have very similar results: the overall correlation between all datasets of these two systems is 0.98. One of the reasons for these results is that the alignment is compromised by the length of the sentences. In the MSRvid dataset, where the sentences are simple such as "*Someone is drawing*", resulting in a good semantic parsing, a high performance for this metric is achieved. However, in the SMT datasets, sentences are much longer (and often ungrammatical, since they are produced by a machine translation system) and the performance of the metric drops. In addition, the alignment makes mistakes such as judging as highly similar sentences such as "*A man is peeling a potato*" and "*A man is slicing a potato*", where the arguments are the same but the situations are different.

4.3 Multi-task Learning-based Semantic Textual Similarity

We propose to model STS by using MTL in order to address challenges observed in previous work. We use a state-of-the-art STS feature set (Šarić et al., 2012) to train an MTL algorithm. As MTL algorithm we use a non-parametric Bayesian model called Gaussian Processes (GP) (Rasmussen and Williams, 2005). We show that the MTL model outperforms previous work on the 2012 datasets, and it has a robust performance on the 2013 datasets. On the STS 2014 challenge our method shows competitive results. However, the challenge of unknown tasks is a sensitive variable that affects the overall performance. In addition, we compare our proposed MTL models with a task adaptation baseline based on a Transductive Support Vector Machine algorithm in terms of unknown tasks.

4.3.1 TakeLab Features Description

We use the features from one the top best performing systems on the STS. The TakeLab[5] system for STS 2012 is publicly available and it extracts the following types of features:

N-gram overlap is the harmonic mean of the degree of matching between the first and second texts, and vice-versa. The overlap is computed for unigrams, bigrams, and trigrams.

[5]http://takelab.fer.hr/sts/

WordNet-augmented word overlap is the partial WordNet path length similarity score assigned to words that are not common to both texts.

Vector space sentence similarity is the representation of each text as a distributional vector by summing the distributional (i.e., LSA) vectors of each word in the text and taking the cosine distance between these texts vectors.

Shallow NE similarity is the matching between named entities that indicates whether they were found in both of the two texts.

Numbers overlap is an heuristic that penalises differences between numbers in texts.

The features make of a vector of 21 similarity scores.

4.3.2 Multi-task Gaussian Process

Gaussian Processes (Rasmussen and Williams, 2005) is a Bayesian non-parametric machine learning framework based on kernels for regression and classification. In GP regression, for the inputs x we want to learn a function f that is inferred from a GP prior:

$$f(x) \sim GP(m(x), k(x, x\prime)), \tag{4.3}$$

where $m(x)$ defines a 0 mean and $k(x, x\prime)$ defines the covariance or kernel functions. In the single output case, the random variables are associated to a process f evaluated at different values of the input x while in the multiple

output case, the random variables are associated to different processes and evaluated at different values of x.

We are interested in the intrinsic coregionalization model for GP. The coregionalization model is an *heterotopic* MTL in which each output is associated with a different set of inputs. In our case the different inputs are the STS tasks. The intrinsic coregionalization model (i.e. MTL-GP) is based on a separable multi-task kernel (Álvarez et al., 2012) of the form:

$$\begin{aligned} \mathbf{K}(\mathbf{X}, \mathbf{X}) &= \mathbf{B} \otimes k(\mathbf{X}, \mathbf{X}) \\ &= \left(B_{ij} \times k(\mathbf{x}_i, \mathbf{x}_j)\right), \end{aligned} \tag{4.4}$$

where $k(X, X)$ is a standard kernel over the input points and B is a positive semi-definite matrix encoding task covariances called coregionalization matrix. B is built from other matrices $B = WW^{\mathsf{T}} + diag(k)$, where W is a matrix that determines the correlations between the different outputs and k is a vector which allows the outputs (i.e. tasks) to behave independently. The representation of data points is augmented with a task id and given the id of a pair of data points the covariance from the standard kernel between points is multiplied by a corresponding covariance from B, which modifies the covariance of the data points depending if they belong to the same task or different tasks.

The coregionalization matrix B controls the amount of inter and intra task transfer of learning among tasks. Cohn and Specia (2013) propose different types of B matrices for predicting the quality of machine translations. They

developed B matrices that represent an explicit intra-task transfer to be a part of the parametrised kernel function. However, we use a default B where the weights of the matrix are learnt along with the hyper-parameters by the GP tool.

For training our method we use GPy [6]. We use the combination of the RBF kernel with the coregionalization kernel. The parameters used to build the coregionalization matrix are the number of outputs to coregionalize and the rank of W. For example, the number of outputs to coregionalize are the 3 tasks from the STS 2012 training dataset. The B matrix and the RBF kernel hyper-parameters are jointly optimised by GPy. Each instance of the training data is augmented with the id of their corresponding task. During testing a new instance to be predicted has to be matched to a specific task id from the training data. In the case of an unknown test task we match it to a similar training task, given the task description of the test dataset.

4.3.2.1 Linear Combination of Coregionalization Kernels

In order to cope with unknown test tasks, as a first approach, we add an extra training task into the MTL-GP which consists of all the training instances. The intuition is that the kernel function will assign a higher correlation (i.e. covariance) of the unknown test inputs to similar training inputs (e.g. related tasks) and a lower correlation otherwise. However, the inputs with a low

[6] https://github.com/SheffieldML/GPy

correlation may introduce prediction errors. We select the extra training task for unknown test tasks.

In addition, we use a linear combination of kernels to allow learning differentiation between tasks. In this way, we can control the inter-intra task transfer by choosing which kernel to use for each task. We used a kernel for general learning and task-specific kernels. The resulting kernel has the form:

$$\mathbf{K}_{LCM1}(\mathbf{X},\mathbf{X}) = \mathbf{K}_0(\mathbf{X},\mathbf{X}) + I_{\{task_1\}}\mathbf{K}_1(\mathbf{X},\mathbf{X}) + ... + I_{\{task_D\}}\mathbf{K}_D(\mathbf{X},\mathbf{X}), \quad (4.5)$$

where $\mathbf{K}_{LCM1}(\mathbf{X},\mathbf{X})$ is the sum of separable multi-task kernels, and $I_{\{\cdot\}}$ are index functions that have value 0 or 1 depending on the task. Hence, $\mathbf{K}_0(\mathbf{X},\mathbf{X})$ is the only kernel used for all the tasks, while kernels $\mathbf{K}_i(\mathbf{X},\mathbf{X})$ are private for each task. In the case of the 2012 training set, the kernel $\mathbf{K}_{LCM1}(\mathbf{X},\mathbf{X})$ has one general kernel and three specific kernels. Therefore, any additional task added after training will use only the general kernel. We also set $\mathbf{B}_{i,j} = 1$ for $i = j$ (i.e. same tasks in the coregionalization matrix) and 0 otherwise in all the kernels. Such condition is necessary for the $\mathbf{K}_i$ kernels to be private and allow using $\mathbf{K}_0$ with unseen tasks.

Following the intuition of the combination of kernels we also define the kernel:

$$\mathbf{K}_{LCM2}(\mathbf{X},\mathbf{X}) = \mathbf{K}_I(\mathbf{X},\mathbf{X}) + I_{\{trained\}}\mathbf{K}_{II}(\mathbf{X},\mathbf{X}), \quad (4.6)$$

where $\mathbf{K}_I(\mathbf{X}, \mathbf{X})$ has a coreginalization matrix such that $\mathbf{B}_{i,j} = 0$ for $i \neq j$. $\mathbf{K}_{II}$ has no restrictions, but it is only used for tasks that have been trained.

Furthermore, given the large amount of training data used for the STS 2014 (i.e. 6832 training instances) we also train a sparse GP model from GPy. The main limitation of the GP model is the that memory demands grow $O(n^2)$ and the computational demands grow $O(n^3)$, where n is the number of training instances. Sparse methods for example in (Titsias, 2009) try to overcome this limitation by constructing an approximation on a small set of m support or inducing instances that allow the reduction of computational demands to $O(nm^2)$. For the sparse version on GPy we use the same combination of kernels, we select empirically the number of inducing instances and the GP tool randomly pick the instances from the training data.

4.3.3 Transductive Support Vector Machine

Our main motivation is to use the TSVM as a task adaptation baseline. TSVM takes into consideration a particular test set and tries to minimise errors only on those particular instances (Vapnik, 1995). The particular test set is added into the training dataset without labels. The TSVM learns a large margin hyperplane classifier using labelled training data, but at the same time it forces that hyperplane to be far from the unlabelled data. The TSVM considers f that maps inputs x to outputs y. However, TSVM does not construct a function f where the output of the transduction algorithm

is a vector of labels, and the method transfers the information from labelled instances to the unlabelled.

We use SVM-light Joachims (1998) for training. Our TSVM uses an RBF kernel with no hyper-parameter optimisation. Each unknown task in the test dataset is added (without labels) to the training dataset. This improved training data is used to perform testing.

In the following section we show a comparison with previous work for the STS 2012 and 2013 datasets and the official results for English and Spanish STS 2014 datasets.

4.3.4 Results and Discussion

To evaluate our proposed models we define the following runs[7]:

MTL-GP model based on the multi-task kernel from equation (4.4). In this model each input of the training data is augmented with the index of their corresponding task. During testing a training task with an specific index has to be selected for a new test input. In the case of an unknown test task we set a training task with a similar one, given the description of the task of the test dataset.

MTL-GP extra model based on the multi-task kernel from equation (4.4) with an extra training task to be selected in the case of unknown tests tasks.

[7]We use a sparse GP approximation with m=50 inducing points and with a combination of RBF and coregionalization kernels.

MTL-GP LCM1 model based on the linear combination of multi-task kernels that are the general kernel and specific task kernels from equation (4.5) to avoid the necessity of selecting a training task during testing.

MTL-GP LCM2 model based on the linear combination of the independent and shared multi-task kernels from equation (4.6).

We use the STS datasets, described in Section 4.2.2, to compare our proposed models with previous work. For training we use the STS 2012 training datasets and we compare the results on the STS 2012 with publicly available systems and with the official Baseline based on the cosine metric computed over word overlaps, where the official score is the Pearson correlation. We match the unknown task OnWN to MSRpar given that the task of paraphrasing is news from the web that contains a broad vocabulary. We compare the MTL-GP to 2012, 2013 and 2014 and the TSVM only with the unknown tasks of SMTnews and OnWN in 2012, and the STS 2013.

Table 4.9: Comparison with previous work on the STS 2012 test datasets

Method	MSRpar	MSRvid	SMTeuroparl	SMTnews	OnWN
Šarić et al. (2012)	**0.7343**	0.8803	0.4771	0.3989	0.6797
Bär et al. (2012)	0.68	0.8739	0.5280	0.4937	0.6641
MTL-GP	0.7324	**0.8877**	**0.5615**	0.6053	0.7256
MTL-GP extra	0.7176	0.8877	0.5615	0.5128	0.7055
MTL-GP LCM1	0.7148	0.8670	0.5617	**0.6470**	**0.7406**
MTL-GP LCM2	0.7148	0.8565	0.5526	0.6427	0.7288
TSVM	-	-	-	0.4411	0.6840
Baseline	0.4334	0.2996	0.4542	0.3908	0.5864

Table 4.10: Matching of new 2013 tasks with 2012 training data for the MTL-GP

Test \ Train	MSRVid	MSRpar	SMTeuroparl
Headlines	0.6666	0.6595	0.5693
OnWN	0.6516	0.4635	0.4113
FNWN	0.4062	0.3217	0.2344

Table 4.9 shows the comparison of the MTL-GP with previous work, where our method outperforms in most of the tasks. Our method improves the results of TakeLab that is trained with a separate Support Vector Regression model for each task. We compare our method with the simple version of TakeLab that is available, however there is a different version with syntactic features where the results do not show a significant variation only in SMTnews r=0.4683. For the complete alternative results we refer the reader to (Šarić et al., 2012). The MTL-GP extra, LCM1 and LCM2 achieve closer results to the MTL-GP without the necessity of selecting a task during testing. However, the MTL-GP extra imposes the problem of a slow training because of the extra task, where we have repeated training instances. The model LMC1 outperform the best system on unknown tasks and the model run1.

On the STS 2013 dataset we compare our method with work based on task adaptation and the official baseline. We use the 2012 data for training because of the lack of training data for this dataset. To compare the MTL-GP extra, LCM1 and LCM2 we use the MTL-GP with the best result (MSRvid). Table 4.10 shows all the possible matching combinations between

the STS 2013 test and STS 2012 training data. It is worth mentioning that the best results are given by the matching unseen tasks with the MSRvid task, where all the tasks achieve their highest result with the same training task. Our TSVM baseline is close to the official baseline and to the worst MTL-GP matching. Transductive Learning (TL) transfers the information from labelled instances to the unlabelled by adding a particular test dataset/-task into the training to minimise errors only on those particular instances. Furthermore, the TSVM shows poor results on new tasks, where a possible reason for this performance is the lack of hyper-parameter optimisation.

Table 4.11: Comparison of the best matching MTL-GP (MSRvid) with previous work on STS 2013 test datasets

Method	Headlines	OnWN	FNWN
Heilman and Madnani (2013)	**0.7601**	0.4631	0.3516
Severyn et al. (2013)	0.7465	0.5572	0.3875
MTL-GP	0.6666	**0.6516**	**0.4062**
MTL-GP extra	0.6417	0.6498	0.4036
MTL-GP LCM1	0.6676	0.6163	0.4020
MTL-GP LCM2	0.6615	0.4030	0.4084
TSVM	0.5857	0.61	0.21
Baseline	0.5399	0.2828	0.2146

In Table 4.11, we show the comparison with previous work on the 2013 datasets. Our method shows very competitive results but with the correct matching of task (MSRvid), whereas the worst performed matching (SMTeuroparl, Table 4.10) shows results that are closer to the official baseline. In previous work the task adaptation is performed with the addition of extra features and the subsequent extra parameters to the model, where in

the MTL-GP the transfer learning is done with the coregionalization matrix and does not depend in large amounts of data as opposed to previous work. The MTL-GP extra, LCM1 and LCM2 show comparable results to the best selection of test task for the MTL-GP.

For our official participation[8] the training dataset consists of the combination of each English training and test STS tasks from 2012 and 2013 which gives a number of 7 tasks. During testing on the first run we matched similar datasets/tasks with each other and the new tasks with the MSRpar. For the second run we matched the new task datasets with a related one. The dataset matching (test/training) is as follows: deft-forum/MSRpar, deft-news/SMTnews, tweet-news/SMTnews and images/MSRvid. For the third run the difference in matching is for deft-news/headlines and tweet-news/headlines, where the other tasks remain with the same matching. Table 4.12 show the official STS 2014 results where our best method (i.e. run3) achieves the rank 10.

Table 4.12: Official English STS 2014 test datasets results for the MTL-GP

Run	deft-forum	deft-news	headlines	images	OnWN	tweet-news	rank
UoW run1	0.3419	0.7512	0.7535	0.7763	0.7990	0.7368	11
UoW run2	0.3419	0.5875	0.7535	0.7877	0.7990	0.6281	17
UoW run3	0.3419	0.7634	0.7535	0.7877	0.7990	0.7529	10

In Table 4.13, we show the comparison of the MTL-GP and the sparse MTL-GP with the best 2014 system (DLSCU run2). For both MTL methods

[8]We participate with the system called UoW, which is based on the MTL-GP

we match the 2014 tasks with the best scored training task headlines. For the sparse MTL-GP run1 (Table 4.13) we chose experimentally a number of m=500 random induced points. As a reference, the correlation of sparse MTL-GP with 50 points on deft-forum is r=0.4691 on 0.23 hours, 100 points is r=0.4895, with 500 points r=49124 and 1000 points r=0.49108. The sparse MTL-GP with 500 points runs in 1.38 hours compared with 2.39 hours for the full MTL-GP[9]. The sparse version achieves results similar to the full model and it shows very competitive performance compared with the best system. Model run3 shows competitive results compared to the best system in few tasks. However, in the OnWN dataset the run3 shows poor results despite the presence of this task within the training tasks.

Table 4.13: Comparison of the best matching MTL-GP (headlines), Sparse MTL-GP and best system in STS 2014 test datasets

Run	deft-forum	deft-news	headlines	images	OnWN	tweet-news
DLSCU run2	0.4828	**0.7657**	**0.7646**	**0.8214**	**0.8589**	0.7639
MTL-GP	0.4903	0.7633	0.7535	0.8063	0.7222	0.7528
Sparse						
MTL-GP run1	0.4910	0.7642	0.7540	0.8057	0.7276	0.7539
MTL-GP extra	0.4149	0.7530	0.7429	0.7832	0.7793	0.7126
MTL-GP LCM1	**0.4937**	0.7186	0.7282	0.7886	0.6875	**0.7645**
MTL-GP LCM2	0.4089	0.6449	0.7510	0.7331	0.6768	0.7481

For the Spanish STS we use both simple and state-of-the-art (SoA) features to train the MTL-GP. The simple features are similarity scores from string metrics such as Levenshtein, Gotoh, Jaro, etc[10]. The SoA similar-

[9]Intel Xeon(R) at 2.67GHz with 24 cores
[10]https://github.com/Simmetrics/simmetrics

ity scores features come from one of the top ranked STS systems TakeLab. Moreover, the training dataset consists of the combination of each English STS tasks from 2012 and 2013 and the Spanish trial dataset with task-id matching each instance to a given task. We represent the feature vectors with sparse features for the English and Spanish training datasets, where in English the pairs have simple and SoA features and for Spanish only the simple features. In other words, the feature vectors have the same amount of 34 features: 13 simple features and 21 SoA features. However, for Spanish the SoA features are set to 0 in training and testing. The motivation to use SoA and simple features in English is that the extra information will improve the transfer learning on the English task and discriminate between the English tasks and the Spanish tasks, which only contains simple features. For testing we only extract the simple features and the SoA features are set to 0. For the coregionalization matrix we set the number of taks to be the English STS tasks from 2012 and 2013 plus the Spanish trial, where the Spanish is treated as an additional *task*, which results in 8 tasks. In the first run of testing we matched the test datasets to the Spanish task, and for the second run we matched the datasets to the MSRpar task. Table 4.14 shows the official results for the Spanish subtask. We compare our method with the best ranked system, where our method shows competitive performance placed in rank 7. However, we only show the results of run1 because both runs achieved the same performance on the official results.

Table 4.14: Official Spanish STS 2014 test datasets results

Run	Wikipedia	News	Weighted mean	rank
UMCC_DLSI run2	0.78021	0.82539	0.80718	1
UoW run1	0.7483	0.8001	0.7792	7

Table 4.15: Comparison of best system against sparse MTL-GP Spanish STS 2014 results

Run	Wikipedia	News
UMCC_DLSI run2	0.78021	0.82539
Sparse MTL-GP run1	0.7468	0.7959
Sparse MTL-GP run2	0.7380	0.7878

Table 4.15 shows the comparison of the best STS 2014 system against two different sparse MTL-GP matched with the Spanish trial with 500 induced random points. The Sparse MTL-GP run1 uses the sparse features described above and the run2 uses a modification of the feature set that consists of specific features for each type of tasks. In other words, for the English tasks the simple features are set to 0 and for Spanish the SoA are still set to 0. The difference between both sparse MTL-GP models is very low, where the use of all the features on the English tasks improve the results. However, the performance of both models is still lower than the best system.

4.4 Summary

We use our RTE proposed alignment methods for STS and MT evaluation. For MT evaluation the combination of our method with BLEU improves the overall performance. However, the contribution of the alignment method for STS is poor in comparison with simple similarity metrics. Moreover, most

of the methods for STS show good results on some datasets and poor on others. In order to solve this limitation, we propose a system based on MTL along with state-of-the-art features. Our method improves the performance of the state-of-the-art features on the same datasets. Our method shows a competitive performance with methods based on task adaptation.

Chapter 5

Conclusions

In this thesis we have introduced new methods for two types of semantic relations, namely RTE and STS. The methods are based on semantic information from the input texts, and on ML modelling to identify how their inputs and corresponding outputs are related. In this Chapter we revisit our main contributions to the two semantic relations. We also discuss future research directions for our work on both types of relations.

5.1 Contributions Revisited

The main contributions of this work are related with the following research questions:

Recognising Textual Entailment *To what extent the relational information extracted from semantically aligned T-H pairs affects the performance of an RTE method?*

In Chapter 3, we introduced an RTE method employing a multi-stage architecture. In the first stage, we proposed new predicate-argument alignment methods, and for the entailment decision stage we explored the use of different learning models.

The propositional learning model for RTE is based on a new predicate-argument alignment and simple string-based metrics. The method has comparable performance with the average of methods in the RTE Challenges, but is far from the best system. The evaluation of the RTE datasets shows that the coverage of the predicate-argument alignment methods affect its overall performance. The main difference with respect to previous work (Burchardt et al., 2007) is that our method relies on semantically-oriented features only (i.e. predicate-argument information). Our contributions include the method based on measuring the semantic information between the T-H pairs, and the encoding of this semantic information within the entailment decision. Previous work (e.g. (Burchardt et al., 2007)) show that predicate-argument information is relevant for modelling RTE. Our method is able to address T-H pairs for which the presence of predicate-argument information is critical to decide the entailment relation.

The RTE method based on statistical relational learning led to promising results. The main source of errors was still found to be the alignment stage, which has a low coverage. We showed that when an alignment is found, the relational features improve the final entailment decision. However, the objective of the relational features is to improve the final decision with the given alignment structure by adding information into the model. The novelties with respect to previous work (Garrette et al., 2011) are that our method does not rely on a manually set

threshold to decide the entailment relation, and that the source of information used to train the MLN model comes from the RTE datasets, as opposed to ontologies.

Semantic Textual Similarity *To what extent the simultaneous learning of multiple related tasks affects the overall performance of an STS method?*

In Chapter 4 we showed that MTL improves the results of one of the best STS systems, TakeLab. The matching of a training task with a new unknown task during testing is a key variable that affects the overall performance. Our method tends to achieve best results when known/unknown tasks are matched to similar training task (i.e. MSR-par for 2013 and headlines for 2014). However, this is an artificial setting for truly unknown test tasks, given that their identity will not be given for the matching. We show that the proposed linear combination of kernels achieves comparable results to the best MTL-GP model without the limitation of having to know and set a test task manually. In the Spanish subtask, we train our method with English datasets and the Spanish trial data as an additional task. For this subtask our method also achieved competitive results. The novelty with respect to previous work is that our method uses a linear combination of kernels to predict all the STS tasks, as opposed to task/domain adaptation techniques or meta-classification. The MTL model learns a different set of hyperparameters (i.e. coregionalization matrix) by using a gen-

eral or task-specific kernels. The general kernel allows us to leverage information from all the training tasks, which alleviates the problem of poor generalisation on unseen test tasks.

The tasks of RTE and STS aim to generalise semantic needs across NLP applications. Methods based on semantic information and ML modelling contribute to different aspects of RTE and STS. The results vary drastically from task to task even though in theory the data belong to the same task (e.g. MT evaluation datasets for STS and RTE). In practice methods for measuring semantic similarity are designed to fit a particular task in terms of their features and ML modelling. Moreover, even within a particular task, the performance of methods is affected if datasets belong to different tasks. Task-dependent methods are based on the assumption that the test and training datasets are drawn from the same distribution, but in practice training instances are scarce and the test data can thus belong to very different tasks. We show that the use of MTL techniques alleviate the problem of task adaptation. With minimum reformulation, our methods are also applicable to tasks beyond RTE and STS challenges where measuring semantic similarity is needed. For example, in MT evaluation, we show that the addition of our alignment method to a common evaluation metric improves the overall correlation with human judgements. We believe this also applies to many other tasks, such as text summaries generation or evaluation.

5.2 Future Work

Future work on the RTE statistical relational learning model will include improvements in the alignment stage as well as the incorporation of a more robust set relational features, such as using syntactic structures along with the semantic structures into a combined relational model. In other words, it will use different types of alignments (e.g. monolingual word alignment, syntactic alignment, predicate-argument alignment), where the objective of the MLN formulas will be to penalise or reward decisions made by these different aligners. We could also define formulas that relate decisions across aligners.

Future work on STS involves studying the impact of different types of kernel combinations on the overall performance. Another direction is that of deep learning for the task. Most methods for STS use vector space models only as features that are extracted from a preprocessing stage (Bär et al., 2012). These type of models are tools to represent text as continuous vectors of features. The Compositional Distributional Semantics (CDS) theory aims to obtain distributional meaning for sequences of text by composing the continuous vectors into sequences. As a new research direction we can encode this type of representations directly into a GP model, where we can frame the STS as a deep learning problem. The motivation to use deep GP (Damianou and Lawrence, 2013) is to have a model that does not depend on pipeline architectures, where errors made at one stage (e.g. preprocessing)

are propagated through the following stages. A possible deep GP model for STS may consists of two layers. In the first layer the model learns the CDS function which maps a sentence into a feature space (i.e. latent variables). In the second layer the model learns the function which maps the feature vectors to similarity scores. We can start by first modelling the CDS function as a syntactic feature vector (Zanzotto and Dell'Arciprete, 2013) and then expand horizontally the deep GP with other types of vector representations such as syntactic and semantic dependencies.

Bibliography

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., and Wiebe, J. (2014). Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland.

Agirre, E., Cer, D., Diab, M., Gonzalez-agirre, A., and Guo, W. (2013). sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*.

Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 385–393, Stroudsburg, PA, USA.

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266.

Andreevskaia, A., Li, Z., and Bergler, S. (2005). Can shallow predicate argument structures determine entailment? In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

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

Back, T., Fogel, D. B., and Michalewicz, Z., editors (1999). *Evolutionary Computation 1, Basic Algorithms and Operators*. IOP Publishing Ltd., Bristol, UK, 1st edition.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90, Morristown, NJ, USA.

Banea, C., Hassan, S., Mohler, M., and Mihalcea, R. (2012). Unt: A supervised synergistic approach to semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pages 635–642.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Bär, D., Biemann, C., Gurevych, I., and Zesch, T. (2012). Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the First Joint Conference on Lexical and*

Computational Semantics, SemEval '12, pages 435–440, Stroudsburg, PA, USA.

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

Bar-Haim, R., Dagan, I., Greental, I., Szpektor, I., and Friedman, M. (2007). Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 131–136, Prague.

Bayer, S., Burger, J., Ferro, L., Henderson, J., and Yeh, A. (2005). Mitre's submissions to the eu pascal rte challenge. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.*

Bayer, S., Burger, J., Greiff, W., and Wellner, B. (2004). Association for computational linguistics the mitre logical form generation system.

Beltagy, I., Chau, C., Boleda, G., Garrette, D., Erk, K., and Mooney, R. (2013). 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*, pages 11–21. Atlanta, Georgia, USA.

Bos, J. and Markert, K. (2005). Combining shallow and deep nlp methods for recognizing textual entailment. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Bos, J. and Markert, K. (2006). When logical inference helps determining textual entailment (and when it doesnt). In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Burchardt, A. and Frank, A. (2006). Approaching textual entailment with lfg and framenet frames. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Burchardt, A., Reiter, N., Thater, S., and Frank, A. (2007). A semantic approach to textual entailment: System evaluation and task analysis. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 10–15, Prague.

Cabrio, E., Kouylekov, M., and Magnini, B. (2008). Combining specialized entailment engines for rte-4. In *Proceedings of the Text Analysis Conference 2008,* Gaithersburg, Md., 17–19 November 2008.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O. (2010). Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of*

the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pages 17–53, Uppsala, Sweden.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada.

Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland.

Castillo, J. and Estrella, P. (2012). Semantic textual similarity for mt evaluation. In *Seventh Workshop on Statistical Machine Translation*.

Castillo, J. J. (2010). Textual entailment search task: An initial approach based on coreference resolution. *Intelligent Computing and Cognitive Informatics, International Conference on*, 0:388–391.

Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M.-C., Ramage, D., Yeh, E., and Manning, C. D. (2007). Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 165–170, Prague.

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

Cohn, T. and Specia, L. (2013). Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *51st Annual Meeting of the Association for Computational Linguistics*, ACL-2013, pages 32–42, Sofia, Bulgaria.

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

Cooper, R., Crouch, R., van Eijck, J., Fox, C., van Genabith, J., Jaspars, J., Kamp, H., Pinkal, M., Milward, D., Poesio, M., Pulman, S., Asher, N., Dekker, P., Konrad, K., Krahmer, E., Maier, H., and Ruhrberg, P. (1996). Fracas: A framework for computational semantics.

Corley, C. and Mihalcea, R. (2005). Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA.

Crouch, R. S. and King, T. H. (2006). Semantics Via F-Structure Rewriting. In Butt, M. and King, T. H., editors, *Lexical Functional Grammar Conference 2006*.

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

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

Dagan, I., Roth, D., Sammons, M., and Zanzotto, F. M. (2013). *Recognizing Textual Entailment: Models and Applications*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Damianou, A. C. and Lawrence, N. D. (2013). Deep gaussian processes. In *AISTATS*, pages 207–215.

Das, D. and Martins, A. F. T. (2007). A survey on automatic text summarization. Technical report, Literature Survey for the Language and Statistics II course at Carnegie Mellon University.

Daumé, III, H., Kumar, A., and Saha, A. (2010). Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 53–59, Stroudsburg, PA, USA.

de Marneffe, M.-C., MacCartney, B., Grenager, T., Cer, D., Rafferty, A., and Manning, C. D. (2006). Learning to distinguish valid textual entail-

ments. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Delmonte, R., Bristot, A., Boniforti, M. A. P., and Tonelli, S. (2006). Coping with semantic uncertainty with venses. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Delmonte, R., Bristot, A., Piccolino Boniforti, M. A., and Tonelli, S. (2007). Entailment and anaphora resolution in rte3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 48–53, Prague.

Delmonte, R., Tonelli, S., Piccolino Boniforti, M. A., Bristot, A., and Pianta, E. (2005). Venses - a linguistically-based system for semantic evaluation. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.*

Denkowski, M. and Lavie, A. (2010). Meteor-next and the meteor paraphrase tables: Improved evaluation support for five target languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 339–342.

Fellbaum, C., editor (1998). *WordNet An Electronic Lexical Database.* The MIT Press, Cambridge, MA ; London.

Fowler, A., Hauser, B., Hodges, D., Niles, I., Novischi, A., and Stephan, J. (2005). Applying cogex to recognize textual entailment. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

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

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

Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E., and Dolan, B. (2008). The fourth pascal recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference 2008,* Gaithersburg, Md., 17–19 November 2008.

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

Giménez, J., Márquez, L., Comelles, E., Castellón, I., and Arranz, V. (2010). Document-level automatic mt evaluation based on discourse representations. In *Proceedings of the Joint Fifth Workshop on Statistical Machine*

Translation and MetricsMATR, WMT '10, pages 333–338, Stroudsburg, PA, USA.

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

Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J. (2013). Umbc ebiquity-core: Semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Atlanta, Georgia, USA.

Harmeling, S. (2007). 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*, pages 137–142, Prague.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

He, Y., Du, J., Way, A., and van Genabith, J. (2010). The dcu dependency-based metric in wmt-metricsmatr 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 349–353, Stroudsburg, PA, USA.

Heilman, M. and Madnani, N. (2012). Ets: Discriminative edit models for paraphrase scoring. In *Proceedings of the First Joint Conference on Lexical*

and Computational Semantics, SemEval '12, pages 529–535, Stroudsburg, PA, USA.

Heilman, M. and Madnani, N. (2013). Henry-core: Domain adaptation and stacking for text similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 96–102, Atlanta, Georgia, USA.

Hirst, G. and St-Onge, D. (1997). Lexical chains as representations of context for the detection and correction of malapropisms.

Inkpen, D., Kipp, D., and Nastase, V. (2006). Machine learning experiments for textual entailment. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Jijkoun, V. and de Rijke, M. (2005). Recognizing textual entailment using lexical similarity. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Jimenez, S., Becerra, C., and Gelbukh, A. (2012). Soft cardinality: A parameterized similarity function for text comparison. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 449–453, Stroudsburg, PA, USA.

Joachims, T. (1998). Making large-scale svm learning practical. LS8-Report 24, Universität Dortmund, LS VIII-Report.

Jong, K. A. D. and Spears, W. M. (1990). An analysis of the interacting roles of population size and crossover in genetic algorithms. In *PPSN*, volume 496 of *Lecture Notes in Computer Science*, pages 38–47.

Kamp, H. and Reyle, U. (1993). From discourse to logic; an introduction to modeltheoretic semantics of natural language, formal logic and drt.

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

Kouylekov, M. and Magnini, B. (2005). Recognizing textual entailment with tree edit distance algorithms. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Kouylekov, M. and Magnini, B. (2006). Tree edit distance for recognizing textual entailment: Estimating the cost of insertion. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA.

Lin, D. (1998a). An Information-Theoretic Definition of Similarity. In Shavlik, J. W. and Shavlik, J. W., editors, *ICML*, pages 296–304. Morgan Kaufmann.

Lin, D. (1998b). 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*, pages 768–774, Montréal, Canada.

Lin, D. and Pantel, P. (2001a). Dirt-discovery of inference rules from text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining(KDD-01)*, pages pp. 323–328.

Lin, D. and Pantel, P. (2001b). Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7(4):343–360.

Litkowski, K. (2006). Componential analysis for recognizing textual entailment. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Liu, C., Dahlmeier, D., and Ng, H. T. (2010). Tesla: translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 354–359, Stroudsburg, PA, USA.

Lloret, E., Ferrández, Ó., Muñoz, R., and Palomar, M. (2008). A text summarization approach under the influence of textual entailment. In *NLPCS*, pages 22–31.

Lo, C.-k. and Wu, D. (2011). Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 220–229, Stroudsburg, PA, USA.

MacCartney, B., Galley, M., and Manning, C. D. (2008). A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811, Honolulu, Hawaii.

MacCartney, B., Grenager, T., de Marneffe, M.-C., Cer, D., and Manning, C. D. (2006). Learning to recognize features of valid textual entailments. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 41–48, New York City, USA.

MacCartney, B. and Manning, C. D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague.

Machacek, M. and Bojar, O. (2014). Results of the wmt14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Trans-*

lation, pages 293–301, Baltimore, Maryland, USA.

Macháček, M. and Bojar, O. (2013). Results of the WMT13 metrics shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria.

Magnini, B., Zanoli, R., Dagan, I., Eichler, K., Neumann, G., Noh, T.-G., Pado, S., Stern, A., and Levy, O. (2014). The excitement open platform for textual inferences. In *Proceedings of the ACL 2014 System Demonstrations*.

Malakasiotis, P. and Androutsopoulos, I. (2007). Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47, Prague.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Marsi, E., Krahmer, E., and Bosma, W. (2007). Dependency-based paraphrasing for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 83–88, Prague.

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

Mirkin, S., Dagan, I., and Pado, S. (2010). Assessing the role of discourse references in entailment inference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1209–1219, Uppsala, Sweden.

Mitkov, R., Evans, R., Orasan, C., Dornescu, I., and Rios, M. (2012). Coreference resolution: To what extent does it help nlp applications? In *TSD*, pages 16–27.

Moldovan, D. I. and Rus, V. (2001). Logic form transformation of wordnet and its applicability to question answering. In *ACL*, pages 394–401.

Negri, M. and Kouylekov, M. (2009). Question answering over structured data: an entailment-based approach to question analysis. In *RANLP*, pages 305–311.

Nielsen, R. D., Ward, W., and Martin, J. H. (2006). Toward dependency path based entailment. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Noeman, S. (2013). Ibm eg-core: Comparing multiple lexical and ne matching features in measuring semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Atlanta, Georgia, USA.

Padó, S., Cer, D., Galley, M., Jurafsky, D., and Manning, C. D. (2009a). Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*, 23:181–193.

Padó, S., Galley, M., Jurafsky, D., and Manning, C. D. (2009b). Robust machine translation evaluation with entailment features. In *Proceedings of ACL-IJCNLP*, pages 297–305.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA.

Park, E.-K., Ra, D.-Y., and Jang, M.-G. (2005). Techniques for improving web retrieval effectiveness. *Inf. Process. Manage.*, 41(5):1207–1223.

Pazienza, M. T., Pennacchiotti, M., and Massimo Zanzotto, F. (2005). Textual entailment as syntactic graph distance: a rule based and a svm based approach. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pérez, D. and Alfonseca, E. (2005). Application of the bleu algorithm for recognising textual entailments. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

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

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

Rios, M., Aziz, W., and Specia, L. (2012). Uow: Semantically informed text similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 673–678, Montréal, Canada.

Rios, M. and Gelbukh, A. (2012a). Recognizing textual entailment with similarity metrics. *Research in Computing Science*, 58(ISSN 1870-4069):337347.

Rios, M. and Gelbukh, A. F. (2012b). Recognizing textual entailment with a semantic edit distance metric. In *MICAI (Special Sessions)*, pages 15–20. IEEE.

Rios, M. and Specia, L. (2014). Uow: Multi-task learning gaussian process for semantic textual similarity. In *SemEval-2014: Semantic Evaluation Exercises International Workshop on Semantic Evaluation (SemEval-2014) Co-located with COLING and *Sem*, Dublin, Ireland.

Rios, M., Specia, L., Gelbukh, A., and Mitkov, R. (2014). Statistical relational learning to recognise textual entailment. In *Proceedings of the 15th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2014)*.

Roth, D. and Sammons, M. (2007). Semantic and logical inference model for textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 107–112, Prague.

Rus, V., Lintean, M., Banjade, R., Niraula, N., and Dan, S. (2013). SEMILAR: The Semantic Similarity Toolkit. In *51st Annual Meeting of the Association for Computational Linguistics*.

Rus, V., McCarthy, P. M., Lintean, M. C., McNamara, D. S., and Graesser, A. C. (2008). Paraphrase identification with lexico-syntactic graph subsumption. In *FLAIRS Conference*, pages 201–206.

Schuler, K. K. (2006). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania.

Severyn, A., Nicosia, M., and Moschitti, A. (2013). ikernels-core: Tree kernel learning for textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 53–58, Atlanta, Georgia, USA.

Shareghi, E. and Bergler, S. (2013). Clac-core: Exhaustive feature combination for measuring textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Atlanta, Georgia, USA.

Stern, A. and Dagan, I. (2011). A confidence model for syntactically-motivated entailment proofs. In *RANLP*, pages 455–462.

Thomason, R., editor (1974). *The collected papers of Richard Montague*. Yale University Press.

Titsias, M. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Tratz, S. and Hovy, E. (2008). Summarisation evaluation using transformed basic elements. In *Proceedings TAC 2008*.

Vanderwende, L., Coughlin, D., and Dolan, B. (2005). What syntax can contribute in entailment task. In *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Šarić, F., Glavaš, G., Karan, M., Šnajder, J., and Bašić, B. D. (2012). Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 441–448, Stroudsburg, PA, USA.

Wang, H., Rodrguez, S., Dirik, C., Gole, A., Chan, V., and Jacob, B. L. (2005). Terps: the embedded reliable processing system. In *ASP-DAC*, pages 1–2.

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

Wang, M. and Manning, C. D. (2012). Spede: Probabilistic edit distance metrics for mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 76–83, Stroudsburg, PA, USA.

Wang, R. and Neumann, G. (2007). Recognizing textual entailment using a subsequence kernel method. In *In Proceedings of AAAI*.

Wong, B. T.-M. and Kit, C. (2010). The parameter-optimized atec metric for mt evaluation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 360–364, Stroudsburg, PA, USA.

Wu, S., Zhu, D., Carterette, B., and Liu, H. (2013a). Mayoclinicnlpcore: Semantic representations for textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Atlanta, Georgia, USA.

Wu, X., Yu, H., and Liu, Q. (2013b). DCU participation in WMT2013 metrics task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 435–439, Sofia, Bulgaria.

Yeh, E. and Agirre, E. (2012). Sriubc: Simple similarity features for semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 617–623, Stroudsburg, PA, USA.

Zanzotto, F., Moschitti, A., Pennacchiotti, M., and Pazienza, M. (2006). Learning textual entailment from examples. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Zanzotto, F. M. and Dell'Arciprete, L. (2013). Transducing sentences to syntactic feature vectors: an alternative way to "parse"? In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 40–49, Sofia, Bulgaria.

Zanzotto, F. M., Pennacchiotti, M., and Moschitti, A. (2007). Shallow semantic in fast textual entailment rule learners. In *Proceedings of the ACL-*

PASCAL Workshop on Textual Entailment and Paraphrasing, pages 72–77, Prague.