Abstract

We present a system for resolving both semantic relatedness (SR) and textual entailment (TE) tasks. There are two major contributions the method proposed here brings to the field: (1) it shows that there is a correlation between the SR scores and TE judgments which can be used to improve the accuracy of both of these tasks and (2) it shows that we can handle the structural information via patterns extracted from corpora and that this approach brings a substantial improvement to distributional systems. The system attains a new state of the art for TE and reaches a correlation score within 1.5% percent to the SR state of the art.

1 Introduction

In the last years, two tasks that involve meaning processing, namely Semantic Relatedness (SR) (Mareilli et al., 2014b) and Textual Entailment (TE) (Dagan et al., 2006), have received particular attention. In various academic competitions, both the TE and SR tasks have been proposed, and useful benchmarks have been created. Yet, till last year, no corpus annotated with both SR and TE labels was available. In the last SemEval 2014, the SICK corpus containing both SR and TE annotation for the same pairs have been released (Mareilli et al., 2014a). The importance of this corpus comes from the fact that many systems for resolving either one of the tasks are actually quite similar, but the relation between SR and TE has not been analyzed yet. SICK allows a direct comparison between the systems that address both tasks.

Many of these systems exploit the distributional information of words via word based similarity vectorial models. It has been shown that structural methods, such as syntactic tree kernel systems, only improve marginally the overall results. However, the syntactic structure plays an important role in understanding the meaning of a given text, the intuition being that without an appreciation of the syntactic structure of a language, the study of lexical semantics is bound to fail. There is no way in which meaning can be completely divorced from the structure that carries it. (Pustejovsky, 1991)

In this paper we present a system that combines distributional and structural information for both the SR and TE tasks. There are two major contributions the method proposed here brings to the field: (1) it shows that there is a correlation between the SR scores and TE judgments which can be used to improve the accuracy of both of these tasks and (2) it shows that we can handle the structural information via patterns extracted from corpora and that this approach brings a substantial improvement to distributional systems. By employing 1) and 2) we built a system that performs competitively in SR and TE tasks reaching a new state of the art on the SICK corpus for TE and being less than 1.5% below the state of the art for SR.

This paper is organized as follows: In the next section we review the related work. In Section 3, we analyze the relation between SR and TE. In Section 4 we describe the relationship between corpus patterns and meaning processing, with focus on TE. In Section 5 we describe the relation between corpus patterns and textual entailment. The preprocess-
ing required to apply correctly the corpus patterns is described in Section 6. The architecture of the whole system is presented in Section 7. In Section 8 we present the experimental results, comparing our method against a suite of distributional and structural approaches.

2 Related Work

Based on Harris Distributional Hypothesis (HDH), many approaches to Word Sense Disambiguation (WSD) have focused on the contexts formed by the words surrounding the target word. With respect to verb behavior, selectional restrictions have been used in WSD (McCarthy et al., 2001), (Briscoe et al., 2006). Also, (Hindle, 1990) has tried to classify English nouns in similarity classes by using a mutual information measure with respect to the subject and object roles. Such information is very useful only in certain cases and, as such, it is difficult to use it directly in doing WSD. Lin and Pantel (Lin and Pantel, 2001) transpose the HDH from words to dependency trees. However, their measure of similarity is based on a frequency measure. They maintain that a (slotX, he) is less indicative than a (slotX, sheriff). While this might be true in some cases, the measure of similarity is given by the behavior of the other components of the contexts: both he and sheriff act either exactly the same with respect to certain verb meanings, or totally differently with respect to others. However, their method cannot be extended to take into account such differences. A classification of these cases is instrumental for WSD. Equally important thing is to overcome the limitation of considering only the subject and object. It has been shown that closed class categories, especially prepositions and particles, play an important role in disambiguation and wrong predictions are made if they are not taken into account (Collins and Brooks, 1995) (Stetina and Nagao, 1997). In contrast, our approach addresses both issues.

Zhao, Meyers and Grishman (Zhao et al., 2004) proposed a SVM application to slot detection, which combines two different kernels, one of them being defined on dependency trees. Their method tries to identify the possible fillers for an event, but it does not attempt to treat ambiguous cases; also, the matching score algorithm makes no distinction between the importance of the words, considering equal matching score for any word within two levels of the dependency tree. (Kulkarni and Pedersen, 2005) have clustered together the examples that represent similar contexts for WSD. However, given that they adopt mainly the methodology of ordered pairs of bi-grams of substantive words, their technique works only at the word level, which may lead to a data sparseness problem. Ignoring syntactic clues may increase the level of noise, as there is no control over the relevance of a bi-gram. A semi-supervised technique for the discovery of semantic pattern is presented in (Sun and Grishman, 2010). Their paper takes into account only the named entities - person (PER), geopolitical entity (GPE), location (LOC), etc. While the authors tried to catch meaning relations between their patterns, there is no clear meaning associated with each pattern. The semantic binary relations discoverable in text are the focus of the paper (Sun and Grishman, 2010). They individuate syntactico-semantic structures which could be encoded as patterns but they do not discuss the complexity of learning them. The paper does not discuss possible extensions of the presented method to patterns matching a whole sentence.

3 Semantic Relatedness and Textual Similarity and Sentence Structure

The SICK corpus, being annotated with both relatedness and entailment information, is a valuable source of information on the relationship existing between the two tasks. We investigate the distribution of relatedness score with respect to the entailment type on the SICK training data. It turns out that the best trade-off between
large interval dimension and high purity is obtained for $\epsilon = 1/4$. The distribution of entailment types inside the relatedness interval of this length, i.e. 0.25, is plotted in Figure 1.

**Linear separability** The results of the previous analysis show that there is a strong correlation between relatedness and entailment type. Looking to the marginal relatedness scores, such as [1-2] or [4.25-5], one can predict the entailment type on the basis of the relatedness score. And vice versa, which means that we can correct the prediction of one variable by regressing it to the other. We are interested mainly in the correction of the relatedness score via the entailment decision. To this end, we computed the probability of the median of each relatedness score given the entailment.

**Active Learning** The linearity of the dependency between the two scores, gives us a direct pipeline architecture. First we compute the relatedness between two sentences. Secondly, we predict the entailment, based on the probabilities computed on the training corpus. Based on structural information, obtained via corpus patterns, we correct the entailment decision in a third step. Using these new decision as an oracle, at the fourth step, we correct the initial relatedness score as well, by adjusting it to the most probable median of the intervals of length $\epsilon$, see Figure 2. The detailed framework and the values for parameters is detailed in Section 5.

4 Corpus Patterns

In this section we present a methodology for acquiring corpus patterns (CP) centered around verbs. These patterns encode the lexical, syntactic and semantic information needed for meaning processing. By recognizing such patterns in text it is possible to draw sound inferences regarding the similarity or entailment of a pair of verbal phrases. First we analyze a property of natural languages, namely phrase non ambiguity, which leads to a working definition of corpus patterns and secondly we review the CPs acquisition methodology.

4.1 Ambiguous and Non Ambiguous Phrases

Some phrases, just like words, are ambiguous and the senses of their words change according to context. For example, the phrase I see is ambiguous, as at least the sense of the verb see is not clear - it could mean both understand or perceive light. However, by considering a larger context, the phrase may become unambiguous, that is, the senses of the words in the phrases do not change anymore whatever new context is added left or right. We call such phrases sense stable phrases (SSPs). For example, by considering a larger context for I see, as in I see your dog or as in I see your reason, we obtain a SSP.

We are interested especially in minimal SSP, which is the minimally necessary context around the verb that creates a SSP. We will show that words inside an SSPs are characterized by specific relationships, which combine lexical and syntactic information. On the basis of these relationships we can derive an automatic methodology of acquisition and recognition of SSPs. Minimal SSPs correspond to a pattern that combines syntactic roles and ontological traits, called lexical or semantic types (Popescu 2007, Hanks 2005), of the words. For example:

```
sbj=[Human] see #1 obj=[PhysicalEntity]
sbj=[Human] see #3 obj=[Stating]
sbj=[Human] drive #1 prepTO=[Location]
sbj=[Human] drive #5 prepTO=[PsychoState]
```

where sbj, obj, prepTO mark the syntactic function, verbs at #1, #3 and #5 mark the verb sense according to WordNet, and between the square brackets we mark the semantic types as specified in an ontology. We call the above patterns as corpus patterns (CP), because they are derived entirely on the basis of the verb behavior in corpus. Each context matched by a CP is a SSP and this ensures that the meaning necessary information is encoded in the CP,
as any other context does not change the sense of the words.

Due to the above relationships between the senses of the words in a minimal SSP, in order to process the meaning of the verbal phrase we need to find the semantic type of just one of the elements of the corresponding CP. In the next section we use this property to resolve semantic tasks.

4.2 Acquisition of Corpus Patterns

The supervised techniques to cluster corpus examples to obtain candidates of CPs is described in (Popescu and Magnini, 2007), (Kawahara et al., 2014). The basic idea is to cluster corpus examples of verb phrases according to their syntactic similarity. Then, to cluster the words on the same syntactic position according to their lexical property and the verb sense. The same class of words is represented by a semantic type that is taken from an ontology. We use the SUMO (Niles and Pease, 2001) ontology, which is an ontology aligned with WordNet, on examples taken from Semcor.

Another similar approach is to use a statistical approach to extract the CPs from other resources, such as Pdev (Hanks 2010) or OntoNotes (Weischedel et al., 2011), via stochastic finite automata or using Naive Bayesian formula (Popescu 2012), (Popescu et al. 2014). Basically, for each class of verbs as defined in Pdev or VerbNet we construct a confusion matrix for each syntactic slot and SUMO attribute. The posterior probability of a CP is computed from the priors in the confusion matrix. The process is sound, because the CPs have a regular language form, thus there are a finite number of differences between the CPs that are associated with a certain verb.

In both cases, the output is a list of corpus patterns which use WordNet sense and SUMO semantic types. The pseudo code below summarizes the two approaches.

5 Corpus Patterns and Textual Entailment

In this section we analyze the relationship between CP and textual entailment. In (Popescu et al., 2011) it was shown that there is a direct relation between corpus patterns and textual entailment. CPs represent an intermediate level of information representation between text and logical formula, and they encode the necessary contextual information for deciding on the phrase meaning. As such, they can be used as input for a logical decision process. We will show that by matching CPs against a pair of sentences we can decide on the correct entailment relationship between the two sentences. Also, we show that we can manage the relationship between a pair of sentences with different polarity, because CPs can handle the differences between positive vs. negative sentences entailment. It is essentially the same set of conditions, except for the fact that in a negative sentence, the scope of negation is firstly determined and then the entailment decision is inverted according to the logic negation.

5.1 Positive Sentences Entailment via CPs

First let us consider an example of pair of sentences from SICK corpus, neither of which contains negation: (s1) A lemur is biting a person’s finger. vs. (s2) An animal is biting a person’s finger.

In this example of pairs of sentences on which we should make a decision regarding entailment, the same corpus pattern is matched. This ensures that the same sense of the verb is used in both sentences. It is sufficient to observe individual relations between the slots of the CPs, because CPs ensure strict entailment conditions: the first sentence entails the second sentence if the semantic type present on a slot in the first CP is a hyponym of the semantic type of the second CP. In general, CP₁ matching the first sentence and CP₂ matching the second sentence, we have the following schema:

\[ s₁ \rightarrow s₂ \text{ if } \]
\[ \text{CP₁: sbj=ST₁ v₁#n₁ obj=ST₂ pp=ST₃} \rightarrow \]
\[ \text{CP₂: sbj=ST₄ v₂#n₂ obj=ST₅ pp=ST₆} \]
if $v_1#n_1 \rightarrow v_2#n_2$

ST$_1$ → ST$_4$ (ST$_1$ is a hyponym of ST$_4$)
ST$_2$ → ST$_5$ (ST$_2$ is a hyponym of ST$_5$)
ST$_3$ → ST$_6$ (ST$_3$ is a hyponym of ST$_6$)

In the example above, lemur is a type of animal, that is lemur is a hyponym of animal, thus we have entailment. The condition $v_1#n_1 \rightarrow v_2#n_2$ requires that the two main verbs should belong to the same WorNet synset, or to the same VerbNet group, or that we learned from training that the two verbs are synonyms (see Section 6.2). The hyponym condition is strict, if the words are characterized by the same semantic type, then there is no entailment and there is no entailment in the reverse order. In conclusion, matching CPs against two sentences that do not contain negation and checking the rule above ensures that the entailment decision is properly taken.

5.2 Negative Sentences Entailment via CPs

Negation is treated similarly, except that the CONTRADICTION decision is made.

6 Linguistic Preprocessing

The corpus pattern methodology described in the previous two sections takes into account the head of the nominal phrases of sentences in active form. However, there are different linguistic forms which require processing for accurate corpus pattern matching. We have considered in our system two main linguistic levels for preprocessing: syntactic and lexical. At the syntactic level, a sentence may exhibit different forms such as passive, coordination, negation, apposition, relative clauses etc. At the lexical level, the constituents of a sentence may display synonymic and antonymic phrases, adverbial and adjectival modifiers, metonymy, synecdoche, semantic void heads etc. We considered that the scope of negation and of coordination is the whole phrase, hypothesis which is true in more than 99% of the time in the SICK corpus.

6.1 Syntactic Preprocessing

Passive The Stanford parser indicates explicitly the passive form, so we simply map the agent and passive subject to the corresponding subject and direct object of the active form.

Coordination From the output of the Stanford parser we can identify coordination and we consider the cases when the subject, verb or object is multiple. We create a set of corpus patterns for each element of the coordination. For example, the sentence A man, a woman and two girls are walking on the beach is matched by the pattern $\text{subj}=[\text{Human}]$ walk $\text{prepON}=[\text{Location}]$ and we keep a list of three different instances of this pattern with $\text{subj}=[\text{Human}]$ being instantiated by man, woman and girl respectively. For treating modifiers such as a, two, a different module is in charge, see the following subsection, modifiers.

Our system, DSeCP, performed very good even if in some cases, such as coordination, was not the best. In fact, looking to the overall results, Table 2, we see that DSeCP obtained a new state of the art for TE and it got close to the state of the art for SR. Each of the pattern instances are checked for entailment against the pattern of the second sentence in the pair, according to the procedure described in Subsection 5.1.

Negation The two types of negation, existential and factual, are treated according to the rules in Section 5. The scope of negation is considered to be the whole constituent, the subject for the negative existential marker, there is no, and the verbal phrase for the negative factual marker, not.

Apposition or Relative Clause These syntactic constructions bring extra information to the head of the main constituent. This information is recorded for the head and used to decide whether the corresponding slots in the CPs observe the entailment conditions as described in Section 5.1.

6.2 Lexical Preprocessing

While the syntactic constructions presented in the previous subsection are general, so we do not need training to deal with them, the lexical properties of words need to be learned. We used the SICK training to learn the lexical features involved in entailment decision.

The learning procedure works as follows: 1) Consider all the pairs of sentences from the SICK training corpus which have a relatedness score above 4, meaning that they are highly similar and that they are in an entailment relation or contradiction relationship. 2) Identify the CPs for both sentences 3) if
the verb is different or if the elements of the pattern are different, then learn that the respective phrases are synonymic or antonymic, according to the entailment value, entail or contradiction, respectively.

**Supervised Lexical Learning**

Using the technique above, we extracted from the SICK training corpus three types of phrases: synonymic, antonymic and semantically void phrases. The output was:

- **942 pairs of synonymic phrases.** The list contains also metonymic or synecdochtic pairs, like *field* vs. *grass*, but we do not know how many of them there are.
- **47 antonymic phrases**, like *empty* vs. *fill*, or *climb* vs. *get down*, which trigger contradiction between the sentences with no negative markers.
- **6 semantically void phrases**, such as *a group of people*. These types of expressions play an important role in the entailment decision, because instead of the head of such expressions, like *group*, we need to consider the prepositional modifier, *people* in order to have a correct CP match.

**7 The System**

The system is made out of two main modules. In a nutshell, in this architecture the first decision is taken via distributional information both for SR and TE scores, but if structural information is reliably extracted via the CPs then the TE decision is corrected if needed, and then the SR score are corrected as well.

The first module implements a distributional model, called DS, in which the similarity of the sentences is computed on the basis of the word similarity without considering any syntactic information. To each sentence a set of vectors is associated, on the basis of the distance between words, which is computed considering the Lin distance (Lin, 1998) in WordNet (Fellbaum, 1999) by using the WordNet::Similarity toolkit (Pedersen et al., 2004), Wikipedia relatedness (Milne and Witten, 2013), Latent Semantic Analysis (LSA) similarity (Landauer et al., 1998). Another set of vectors are created using topic modeling via Latent Dirichlet Analysis (Blei et al., 2003); we experimented with a number of topics between 80 and 1300. We computed the distance between each vector pair by the means of Cosine similarity\(^1\), Kullback-Leibler\(^2\) and Jensen-Shannon\(^3\) divergences. Finally, we selected 15 features from this system and combined with 18 features extracted by the DKPRO system (Bär et al., 2013), which made a total of 33 features to feed into WEKA machine learning toolkit (Hall et al., 2009) to build a regression model (see Figure 3). The DKPRO system was the best system for SR at Semeval 2012, and was intended to use as a baseline at Semeval2013\(^4\).

On the basis of the similarity score a first entailment decision is made. The rationale for this is that there is a strong correlation between SR score and Entailment decision. On SICK training corpus we computed the following thresholds: for an SR below \(\theta_1 = 3.0\) the entailment probability is less than 10% and for a SR score above \(\theta_2 = 4\) the entailment probability is larger than 90%. This procedure is called SR2TE. (see Figure 1)

The second module consists in implementing the CP methodology presented in Section 5 for entailment decision.

```
parse the sentence
preprocess the syntax forms
if there is a negation mark flagNeg = 1;
if there is a CP matching use the CP
else use the subj, obj and prepositional object with
the most frequent sumo type
preprocess the lexical forms
check the CP entailment hyperony condition for each slot
if positive and flagNeg == 0 then ENTAILMENT
if positive and flagNeg == 1 then CONTRADICTION
if negative then NEUTRAL.
```

Taking into account the correlation between SR score and the TE decision, we regress the SR score into the new SR score when the TE decision was changed by the structural module, see Figure 4. The method described in Section 3, produced the following output for the regression parameters: NEUTRAL range [1-4] increases the SR correlation to 0.726; the ENTAILMENT relation associated with

---

\(^1\)http://en.wikipedia.org/wiki/Cosine_similarity
\(^2\)http://en.wikipedia.org/wiki/Kullback-Leibler_divergence
\(^3\)http://en.wikipedia.org/wiki/Jensen-Shannon_divergence
SR score range [5-5] obtains new SR correlation of 0.75014 and the CONTRADICTION relation associated with SR score range [2.5-4.5] has SR correlation of 0.71801.

8 Experiments

In this section we analyze the performances of the above system on the SICK test corpus. We present the contributions of each of the procedures presented in the previous sections and we compare them against the official baseline, against the distributional module, and against three tree kernel approaches, in order to calculate precisely the improvement brought by considering structural information via corpus patterns. The tree kernel approaches we used are reported to perform efficiently and effectively on processing syntactic trees using three proposed approaches Syntactic Tree Kernel (STK) (Moschitti, 2006), Parse Chunk Matching (PCM) (Galitsky, 2013) and Distributed Tree Kernel (DTK) (Zanzotto and Dell’Arciprete, 2012). All these systems have been trained on the SICK training corpora. For the STK system we also use the MSR paraphrasing corpus (Dolan et al., 2005) in order to enhance the system capability to deal with synonymic phrases. In order to understand the behavior of DS, we also present the results using DKPRO system alone, and one system combining DKPRO and the three kernel approaches all together in the same SVM classifier, DKPRO_TTK.

We start by analyzing the performances of the above systems for each type of sentences presented in Section 6, see Table 1. For SR score the Pearson correlation is reported, and for TE relation the accuracy is reported. The exact real number of pairs in each subset is not known, these sets are compiled by our recognizing preprocessing procedures based on the Stanford parser output. However, a manual sample check suggests that these estimates are pretty good, less than 4% errors in recognizing these cases.

From Table 1 we learn that structural information really helps in improving the accuracy, but it needs to take into account also the semantic types. The tree kernel approaches performed poorly by themselves. However, the pairs involving synonymic and antonymic knowledge are difficult for all systems. For semantic void all systems performed notably better. Apparently simply ignoring this construction improves the results as the distributional models outperformed the structural ones. However, we think this is due to the fact that the corpus is biased, in the sense that the sentences with semantic void elements have many other words in common and are usually in entailment relation, thus a system may look, incorrectly, only to the distributional clues. Another salient thing is the fact that combining DKPRO and three kernel approaches into a single classifier does not look like a good idea, as this system scores even with 8% lower than individual three kernel methods.

In Table 2 BST stands for the best system in Semeval 2014, DS⊖ stands for the distributional module without DKPRO features. The DSeCP and BST have comparable results, within 1.7% margin, DSeCP resolving entailment with more accuracy, but performing below BST for SR score.
The results of these experiments show that exploiting the relationship between SR and TE is beneficial for all types of systems, see for example the improvement obtained by DKPROeCP vs. DKPRO. In Figure 5 we plotted the initial SR accuracy vs. the corrected SR accuracy for the real SR score in [2-4] interval. The corrected SR scores gained an increase of approximately 20% accuracy, having a lower variance.
the ontological equivalences in this case. We plan to work on this direction.

References


Octavian Popescu and Bernardo Magnini. 2007. Sense discriminative patterns for word sense disambiguation. In *SCAR workshop, NODALIDA*.


