A Multi-Layer System for Semantic Textual Similarity

Ngoc Phuoc An Vo[1] and Octavian Popescu[2]
[1]*Xerox Research Centre Europe, Meylan, France*
[2]*IBM T.J.Watson Research, YorkTown, U.S.A.*

Keywords: Machine Learning, Natural Language Processing (NLP), Semantic Textual Similarity (STS).

Abstract: Building a system able to cope with various phenomena which falls under the umbrella of semantic similarity is far from trivial. It is almost always the case that the performances of a system do not vary consistently or predictably from corpora to corpora. We analyzed the source of this variance and found that it is related to the word-pair similarity distribution among the topics in the various corpora. Then we used this insight to construct a 4-module system that would take into consideration not only string and semantic word similarity, but also word alignment and sentence structure. The system consistently achieves an accuracy which is very close to the state of the art, or reaching a new state of the art. The system is based on a multi-layer architecture and is able to deal with heterogeneous corpora which may not have been generated by the same distribution.

1 INTRODUCTION

Exhaustive language models are difficult to build because overcoming the effect of data sparseness requires an infeasible amount of training data. In the task of Semantic Text Similarity (STS)[1], the systems must quantifiably identify the degree of similarity between pairs of short pieces of text, like sentences. On the basis of relatively small training corpora, annotated with a semantic similarity score obtained by averaging the opinions of several annotators, an automatic system may learn to identify classes of sentences which could be treated in the same way, as their meaning is basically the same. It has been shown that good results from STS systems may help to improve the accuracy on related tasks, such as Paraphrasing (Glickman and Dagan, 2004), Textual Entailment (Berant et al., 2012), Question Answering (Surdeanu et al., 2011), etc.

However, building a system able to cope with various phenomena which fall under the umbrella of semantic similarity is far from trivial. Various types of knowledge must be considered when dealing with semantic similarity, and the methodology of linking together different pieces of information is a matter of research. It is almost always the case that the performances of a system do not vary consistently or predictably from corpora to corpora. The STS corpora used in STS competitions, and the task description papers (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015) testify that there is no system that consistently scores the best across corpora, and big variation of system performance may occur.

The contribution of this paper consists of threefold: (1) we first investigate the variation of system performance to alleviate the variances, (2) we propose a multi-layer system to comprehensively handle different linguistic features coming from heterogeneous source of data to predict the semantic similarity scores between texts, and (3) we evaluate the system on all available datasets for the task. To our best knowledge, this is the first attempt to evaluate a system on all datasets in STS task. The goal is to present a STS system able to consistently achieve state of the art, or near state-of-the-art result on all STS datasets from 2012 - 2015.

The heterogeneity of sources considered for these corpora makes it difficult to maintain the hypothesis of the same probability distribution of terms for training and testing, therefore we have to adapt our system to handle this situation, which is better described as a mixture of more or less independent and unknown Gaussians. The system is modular having four principal layers: (i) string similarity, (ii) semantic word similarity, (iii) word alignment, and (iv) structural information. These are combined in order to build a classifier which correspond satisfactorily to our goal. To prove this, we present comparatively the results of

[1]http://ixa2.si.ehu.es/stswiki/index.php/Main_Page

Vo, N. and Popescu, O.
A Multi-Layer System for Semantic Textual Similarity.
DOI: 10.5220/0006045800560067
In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR*, pages 56-67
ISBN: 978-989-758-203-5

our system against the top three results for each year individually.

The paper continues as follows: in the next section we present an extensive literature on semantic similarity which proved instrumental in the building of the actual system. In Section 3 we analyze the variation in system performance for STS. In Section 4 we present the system based on four layers. Section 5 describes the experiment settings and Section 6 presents the evaluation results. The paper ends with a section dedicated to conclusions and further work.

2 RELATED WORK

The Semantic Text Similarity (STS) task has become one of the most popular research topics in NLP. Two main approaches have been widely used for tackling this task, namely Distributional Semantic Models (DSMs) and Knowledge-based similarity approaches.

Distributional Semantic Models (DSMs) is a family of approaches based on the distributional hypothesis (Harris, 1968), according to which the meaning of a word is determined by the set of textual contexts in which it appears. These models represent words as vectors that encode the patterns of co-occurrences of a word with other expressions extracted from a large corpus of language (Sahlgren, 2006; Turney et al., 2010). DSMs are very popular for tasks such as semantic similarity. The different meanings of a word are described in a space and words used in similar contexts are represented by vectors (near) in this space. On the basis of such methods, semantically similar words will appear in points near the (semantic) space. Textual contexts can be defined in different ways, thus giving rise to different semantic spaces.

Knowledge-based similarity approaches quantify the degree to which two words are semantically related using information drawn from semantic networks (Budanitsky and Hirst, 2006). Most of the widely used measures (e.g. Leacock and Chodorow, Wu and Palmer, Lin, and Jiang and Conrath, among others) of this kind have been found to work well on the WordNet taxonomy. All these measures assume as input a pair of concepts, and return a value indicating their semantic similarity. Though these measures have been defined between concepts, they can be adapted into word-to-word similarity metrics by selecting for any given pair of words those two meanings that lead to the highest concept-to-concept similarity.

If we focus on sentence to sentence similarity, three prominent approaches are usually employed. The first approach uses the vector space model (Meadow, 1992) in which each text is represented as a vector (bag-of-words). The similarity between two given texts is computed by different distance/angel measures, like cosine similarity, Euclidean, Jaccard, etc. The second approach assumes that if two sentences are semantically equivalent, we should be able to align their words or expressions. The alignment quality can serve as a similarity measure. This approach typically pairs words from two sentences by maximizing the summation of the word similarity of the resulting pairs (Mihalcea et al., 2006). The last approach employs different measures (like lexical, semantic and syntactic) from several resources as features to build machine learning models for training and testing (Bär et al., 2012; Šarić et al., 2012; Shareghi and Bergler, 2013; Marsi et al., 2013; Vo et al., 2014).

As for the specific case of measuring semantic similarity between two given sentences, the Semantic Textual Similarity (STS) tasks[2] [3] have been officially organized and have received an increasing amount of attention (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015).

The UKP (Bär et al., 2012) was the first-ranked system at STS 2012. This system used a log-linear regression model to combine multiple text similarity measures which range from simple measures (word n-grams or common subsequences) to complex ones (Explicit Semantic Analysis (ESA) vector comparisons (Gabrilovich and Markovitch, 2007), or word similarity using lexical-semantic resources). Beside this, it also used a lexical substitution system and statistical machine translation system to add additional lexemes for alleviating lexical gaps. The final models after the feature selection, consisted of 20 features, out of the possible 300+ features implemented.

By contrast, the best system at STS 2013, UMBC EBIQUITY-CORE (Han et al., 2013), adopted and expanded the alignment approach into "align-and-penalize" by giving penalties to both the words that are poorly aligned and to the alignments causing semantic or syntactic contradictions. At the word level, it used a common Semantic Word Similarity model which is a combination of LSA word similarity and WordNet knowledge.

The DLS@CU (Sultan et al., 2014b) achieved best result at STS 2014. It used the word alignment approach described in the literature (Sultan et al., 2014a), which considered several semantic features, e.g. word similarity, contextual similarity, and alignment sequence. It (Sultan et al., 2015) again achieved the best result as shown at STS 2015 us-

[2]http://www.cs.york.ac.uk/semeval-2012/task6/
[3]http://ixa2.si.ehu.es/sts/

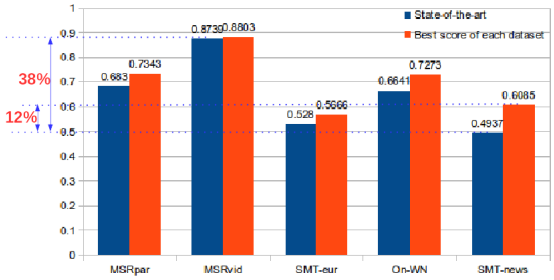

Figure 1: Variance of System Performance in STS 2012.

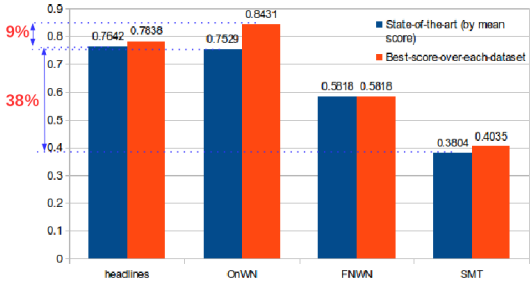

Figure 2: Variance of System Performance in STS 2013.

ing word alignment and similarities between compositional sentence vectors as its features. It adopted the 400-dimensional vectors developed in (Baroni et al., 2014) using the word2vec toolkit (Mikolov et al., 2013) to extract these vectors from a large corpus (about 2.8 billion tokens). Word vectors between the two input sentences were not compared, but a vector representation of each input sentence was constructed using a simple vector composition scheme, then the cosine similarity between the two sentence vectors is computed as the second feature. The vector representing a sentence is the centroid (i.e., the componentwise average) of its content lemma vectors. Finally, these two features are combined using a ridge regression model implemented in scikit-learn (Pedregosa et al., 2011). Besides DLS@CU, it is very interesting that aligning words between sentences has been the most popular approach for other top participants ExBThemis (Hänig et al., 2015), and Samsung (Han et al., 2015).

Besides these approaches, a new semantic representation for lexical was proposed as semantic signature which is the multinomial distribution generated from the random walks over WordNet taxonomy where the set of seed nodes is the set of senses present in the item, (Pilehvar et al., 2013). This representation encompassed both when the item is itself a single sense and when the item is a sense-tagged sentence. This approach was evaluated on three different tasks Textual Similarity, Word Similarity and Sense Similarity; and it also achieved the state of the art on STS 2012 datasets.

3 VARIANCE OF SYSTEM PERFORMANCE IN THE STS TASK

After observing the results from the state-of-the-art systems at the STS 2012 and 2013, we considered one of the biggest problems to address is that results varied from the different corpora, or in other words, the results depend heavily on the given corpora. There are two variances that can be addressed:

- First, the result of the state-of-the-art system is not the best result on each corpus (variance between systems, e.g. state of the art vs best-score system on each corpus).
- Second, the variance of results from the same system on different corpora in Figures 1 and 2 (results varied from 49% to 87% of the state-of-the-art system in STS 2012, and 38% to 76% in STS 2013).

Therefore, we would like to investigate these variances to improve the state of the art and develop a system which can obtain predictable results independently on given corpora.

In this chapter, we analyze the source of variances of accuracy on systems participating in the STS task in 2012 and 2013 by two types of analysis: (1) analysis on the performance of participating systems, and (2) corpora analysis on the various domains of data which affect to the general performance of participating systems.

3.1 Performance Analysis of the STS 2012 - 2013

Firstly we analyze the difference between systems' predictions and gold-standard on each dataset of the STS 2012 and 2013.

Figure 1 shows that there is moderate gap between the performance the of state-of-the-art system and the best-score from other different systems over each corpus. The difference on corpus SMT-news, OnWN and MSRpar are quite large, which are approximately 11%, 6% and 5%, respectively.

Figure 2 shows that there are still some gaps in performance between state of the art and best-score systems on each corpus, except the corpus FNWN, which state of the art system scored highest. Significantly, in the corpus OnWN, the difference is huge, almost 10%.

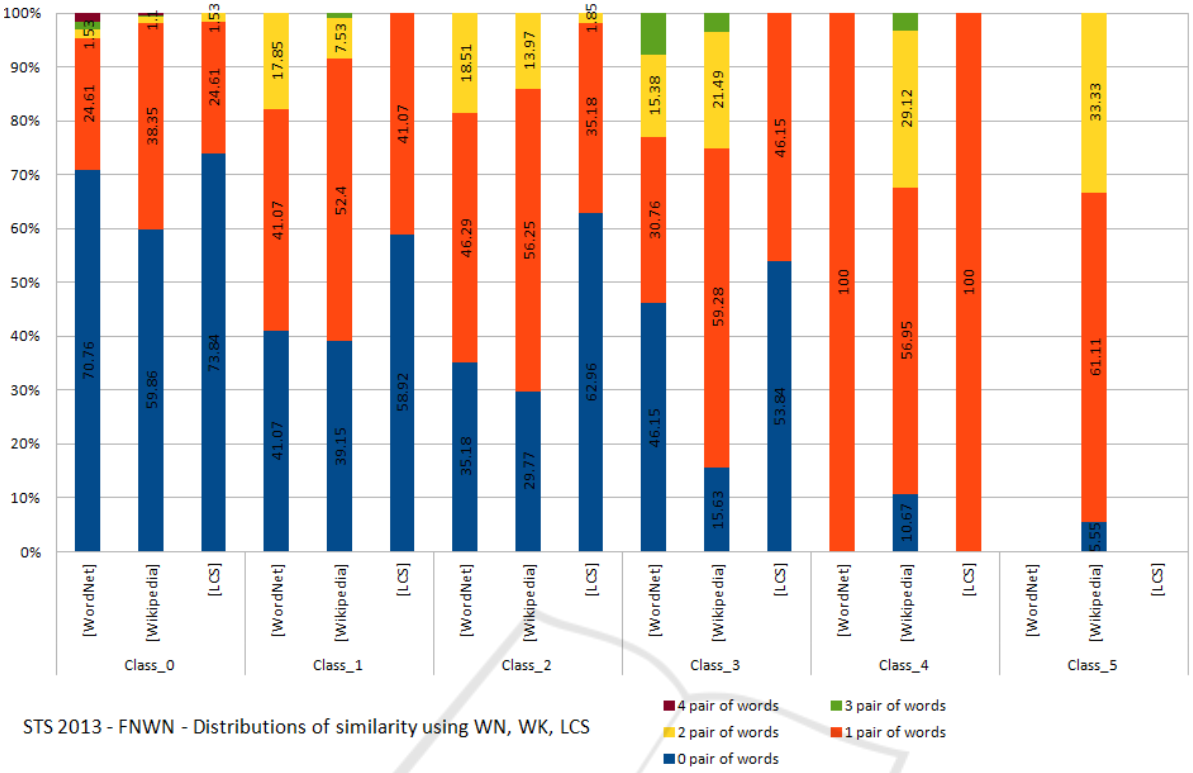

Figure 3: STS 2013 - Corpus FNWN - The word-pair similarity distribution using WordNet, Wikipedia and LCS mapped to the semantic similarity classes [0-5]. Where the classes are gold-standard similarity scores [0-5] classified into different brackets: Class_0 is [0-1), Class_1 is [1-2), Class_2 is [2-3), Class_3 is [3-4), Class_4 is [4-5), and Class_5 is [5].

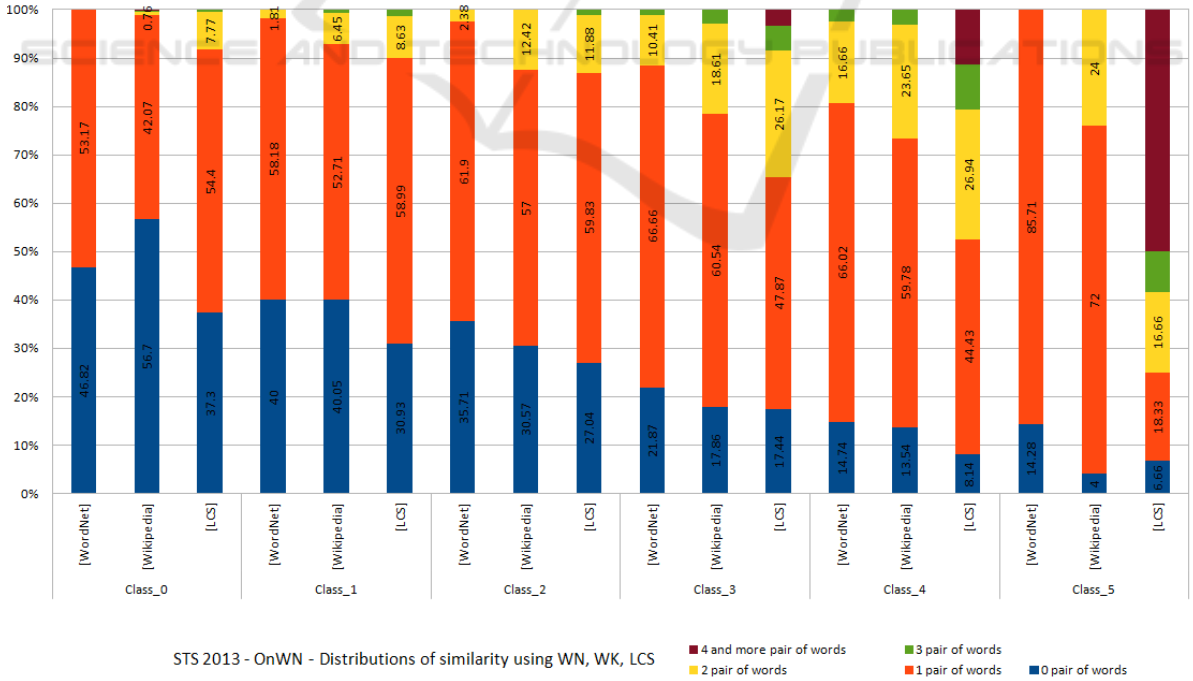

Figure 4: STS 2013 - Corpus ONWN - The word-pair similarity distribution using WordNet, Wikipedia and LCS mapped to the semantic similarity classes [0-5]. Where the classes are gold-standard similarity scores [0-5] classified into different brackets: Class_0 is [0-1), Class_1 is [1-2), Class_2 is [2-3), Class_3 is [3-4), Class_4 is [4-5), and Class_5 is [5].

The heterogeneity of sources makes it difficult for a STS system to score consistently across different corpora. However, the variation observed in the STS tasks is rather significantly big, up to the point that few reliable statements regarding choosing one system over another can be made. In Figures 1 and 2, we plotted the performance of the state-of-the-art system on different datasets at the STS 2012 and 2013. We can see that the accuracy of this system may vary within a window of 38%. This variance is problematic, but another variance is probably more serious than that on a specific corpus, the variance between the best performance of the state-of-the-art system and the performance of best-score system (the best result on different corpora may come from different systems) can be up to 9% - 12%. For the STS task, the margin 9-12% is significant, and many systems achieving results within this distance to the state of the art. The practical question is which system to choose? How can one predict whether one system is really the best system for a new, unknown corpus fed as input?

3.2 Corpora Analysis of the STS 2013

Unless one is able to build systems that cope positively with these variances and the system predictably obtains results within a non significant window to the state of art, the whole approach seems jeopardized. Therefore, it is important to understand the source of this variation and to be able to restrain it within an acceptable margin. In Figures 3 and 4, we plot the distribution of similar word-pairs according to the similarity score. It shows that on the corpora with good results for a simple classifier, there is a good co-variance between word similarity and the similarity scores (Figure 4). Thus, a simple classifier which relies on word and string similarity is more likely to go wrong on the corpus where the similarity score is not necessarily correlated with the number of similar words-pairs.

The second variance shown in Figures 1 and 2 is that the results of the state-of-the-art system are not balanced among the test corpora, and vary from 0.4937 to 0.8739 in STS 2012 and 0.3804 to 0.7642 in STS 2013). In fact, the result of SMT corpus is much lower than others in STS 2013. Most of the systems obtained good results on headlines and OnWN, but very low on FNWN and SMT. It means that most of the systems may learn good features in headlines and OnWN, but not in FNWN and SMT which resulted low scores. In other words, there may be other features remaining in FNWN and SMT that most of systems at the STS 2013 missed. However, it could also be a function of the difficulty of the data.

In order to find a way to alleviate this problem, we investigated the types of similarity existing in the STS 2013 corpora. We used the following common techniques for computing text similarity for our investigation:

- A similarity based on Lin measure (Lin, 1998) using WordNet hierarchy [WN] (computed by the WordNet::Similarity package (Pedersen et al., 2004)).
- A similarity based on Wikipedia concepts [Wiki] (computed by the Wikipedia Miner package (Milne and Witten, 2013)).
- A similarity based on the length of the Longest Common Substring [LCS].

Using these three parameters, we picked ONWN and FNWN datasets[4] for analyzing the number of similar word-pairs between sentence pairs, in accordance to its gold-standard (human annotation) similarity scores in the scale [0-5] split in six classes.

By comparing the plots in Figure 3 (corpus FNWN) vs Figure 4 (corpus OnWN), we can see that the shape of the bars tends to be uniform in Figure 4 while in the Figure 3 the distribution is rather hectic. A threshold separation is likely to work better for corpus ONWN than FNWN. This analysis confirms that the high variance of the system's accuracy is not only related to the word-pair similarity distribution among the gold-standard classes but also other features. In order to improve the accuracy of STS systems, we need to find solutions that add more information on top of the word-pair similarity to improve the separation between classes when the prediction of the word-pair similarity is high.

4 FOUR SEMANTIC LAYERS

In this section we describe our system, which is built from different linguistic features. We construct a pipeline system, in which each component produces different features independently and at the end, all features are consolidated by the machine learning tool WEKA, which learns a regression model for predicting the similarity scores from given sentence-pairs. We adopt few typical STS features in UKP (also known as DKPro) (Bär et al., 2012), such as string similarity, character/word n-grams, and pairwise similarity; however, beyond these typical features, we

[4]http://ixa2.si.ehu.es/sts/index.php%3Foption=com_content&view=article&id=47&Itemid=54.html

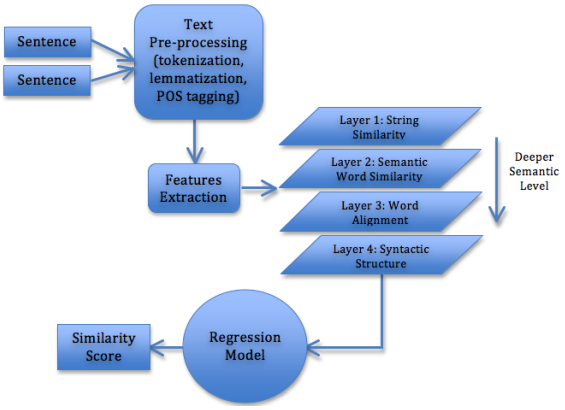

Figure 5: System Overview.

also add other distinguished features, such as syntactic structure information, word alignment and semantic word similarity. The System Overview in Figure 5 shows the logic and design processes in which different components connect and work together.

4.1 Data Preprocessing

The input data undergoes the data preprocessing in which we use Tree Tagger (Schmid, 1994) to perform tokenization, lemmatization, and Part-of-Speech (POS) tagging. On the other hand, we use the Stanford Parser (Klein and Manning, 2003) to obtain the dependency parsing from given sentences.

4.2 Layer One: String Similarity

We use Longest Common Substring (Gusfield, 1997), Longest Common Subsequence (Allison and Dix, 1986) and Greedy String Tiling (Wise, 1996) measures.

Longest Common Substring is the longest string in common between two or more strings. Two given texts are considered similar if they are overlapping/covering each other (e.g sentence 1 covers a part of sentence 2, or otherwise).

Longest Common Subsequence is the problem of finding the longest subsequence common to all sequences in a set of sequences (often just two sequences). It differs from problems of finding common substrings: unlike substrings, subsequences are not required to occupy consecutive positions within the original sequences.

Greedy String Tiling algorithm identifies the longest exact sequence of substrings from the text of the source document and returns the sequence as tiles (i.e., the sequence of substrings) from the source document and the suspicious document. This algorithm was implemented based on running Karp-Rabin matching (Wise, 1993).

4.3 Layer Two: Semantic Word Similarity

Semantic word similarity is the most basic semantic unit which is used for inferring the semantic textual similarity. There are several well-known approaches for computing the pairwise similarity, such as semantic measures using the semantic taxonomy WordNet (Fellbaum, 1998) described by (Leacock et al., 1998; Jiang and Conrath, 1997; Resnik, 1995; Lin, 1998; Hirst and St-Onge, 1998; Wu and Palmer, 1994); or other corpus-based approaches like Latent Semantic Analysis (LSA) (Landauer et al., 1998), Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007), etc.

Among the approaches described above, we deploy three different approaches to compute the semantic word similarity: the pairwise similarity algorithm by Resnik (Resnik, 1995) on WordNet (Fellbaum, 1998), the vector space model Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007), and the Weighted Matrix Factorization (WMF) (Guo and Diab, 2012).

Resnik Algorithm returns a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node). As this similarity measure uses information content, the result is dependent on the corpus used to generate the information content and the specifics of how the information content was created.

Explicit Semantic Analysis (ESA) is a vectorial representation of text (individual words or entire documents) that uses a document corpus as a knowledge base. Specifically, in ESA, a word is represented as a column vector in the TF-IDF matrix (Salton and McGill, 1983) of the text corpus and a document (string of words) is represented as the centroid of the vectors representing its words. The ESA model is constructed by two lexical semantic resources Wikipedia and Wiktionary.[5,6]

Weighted Matrix Factorization (WMF) (Guo and Diab, 2012) is a dimension reduction model to extract nuanced and robust latent vectors for short texts/sentences. To overcome the sparsity problem in short texts/sentences (e.g. 10 words on average), the missing words, a feature that LSA/LDA typically overlooks, is explicitly modeled. We use the pipeline to compute the similarity score between texts.[7]

[5]http://en.wikipedia.org/wiki/Main_Page
[6]http://en.wiktionary.org
[7]http://www.cs.columbia.edu/ weiwei/code.html

Besides these pairwise similarity methods, we also use the n-gram technique at character and word levels. We compare character n-grams (Barrón-Cedeno et al., 2010) with the variance n=2, 3, ..., 15. By contrast, we compare the word n-grams using the Jaccard coefficient done by Lyon (Lyon et al., 2001) and containment measure (Broder, 1997) with the variance of n=1, 2, 3, and 4.

4.4 Layer Three: Word Alignment

At the shallow level of comparing texts and computing their similarity score, we deploy two machine translation evaluation metrics: the METEOR (Banerjee and Lavie, 2005) and TERp (Snover et al., 2006). However, our analysis shows that the TERp result does not really contribute to the overall performance, yet sometimes it may affect our system negatively. Hence, we remove this metric from the system.

Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Banerjee and Lavie, 2005) is an automatic metric for machine translation evaluation, which consists of two major components: a flexible monolingual word aligner and a scorer. For machine translation evaluation, hypothesis sentences are aligned to reference sentences. Alignments are then scored to produce sentence and corpus level scores. We use this word alignment feature to learn the similarity between words and phrases in two given texts in case of different orders.

4.5 Layer Four: Syntactic Structure

Intuitively, the syntactic structure plays an important role for the human being to understand the meaning of a given text. Thus, it also may help to identify the semantic equivalence between two given texts. We exploit the syntactic structure information by the mean of three different approaches: Syntactic Tree Kernel, Distributed Tree Kernel and Syntactic Generalization. We describe how each approach learns and extracts the syntactic structure information from texts to be used in our STS system.

Syntactic Tree Kernel. Given two trees T_1 and T_2, the functionality of tree kernels is to compare two tree structures by computing the number of common substructures between T1 and T2 without explicitly considering the whole fragment space. According to (Moschitti, 2006), there are three types of fragments described as the subtrees (STs), the subset trees (SSTs) and the partial trees (PTs). A subtree (ST) is a node and all its children, but terminals are not STs. A subset tree (SST) is a more general structure since its leaves need not be terminals. The SSTs satisfy the constraint that grammatical rules cannot be broken. When this constraint is relaxed, a more general form of substructures is obtained and defined as partial trees (PTs).

The Syntactic Tree Kernel is a tree kernels approach to learn the syntactic structure from syntactic parsing information, particularly, the Partial Tree (PT) kernel is proposed as a new convolution kernel to fully exploit dependency trees. The evaluation of the common PTs rooted in nodes n_1 and n_2 requires the selection of the shared child subsets of the two nodes, e.g. [S [DT JJ N]] and [S [DT N N]] have [S [N]] (2 times) and [S [DT N]] in common. We use the tool svm-light-tk 1.5 to learn the similarity of syntactic structure.[8]

Syntactic Generalization (SG). Given a pair of parse trees, the Syntactic Generalization (SG) (Galitsky, 2013) finds a set of maximal common subtrees. Though generalization operation is a formal operation on abstract trees, it yields semantics information from commonalities between sentences. Instead of only extracting common keywords from two sentences, the generalization operation produces a syntactic expression. This expression maybe semantically interpreted as a common meaning held by both sentences. This syntactic parse tree generalization learns the semantic information differently from the kernel methods which compute a kernel function between data instances, whereas a kernel function is considered as a similarity measure.

SG uses least general generalization (also called anti-unification) (Plotkin, 1970) to anti-unify texts. Given two terms E_1 and E_2, it produces a more general one E that covers both rather than a more specific one as in unification. Term E is a generalization of E_1 and E_2 if there exist two substitutions σ_1 and σ_2 such that $\sigma_1(E) = E_1$ and $\sigma_2(E) = E_2$. The most specific generalization of E_1 and E_2 is called anti-unifier. Technically, two words of the same Part-of-Speech (POS) may have their generalization which is the same word with POS. If lemmas are different but POS is the same, POS stays in the result. If lemmas are the same but POS is different, lemma stays in the result. The software is available here.[9]

Distributed Tree Kernel (DTK). (Zanzotto and Dell'Arciprete, 2012) This is a tree kernels method using a linear complexity algorithm to compute vectors for trees by embedding feature spaces of tree fragments in low-dimensional spaces. Then a recursive algorithm is proposed with linear complexity to compute reduced vectors for trees. The dot product among reduced vectors is used to approximate the

[8] http://disi.unitn.it/moschitti/SIGIR-tutorial.htm

[9] https://code.google.com/p/relevance-based-on-parse-trees

Table 1: Summary of STS datasets in years 2012 - 2015.

year	dataset	#pairs	source
2012	MSRpar	1500	newswire
2012	MSRvid	1500	video descriptions
2012	OnWN	750	OntoNotes, WordNet glosses
2012	SMTnews	750	Machine Translation evaluation
2012	SMTeuroparl	750	Machine Translation evaluation
2013	headlines	750	newswire headlines
2013	FNWN	189	FrameNet, WordNet glosses
2013	OnWN	561	OntoNotes, WordNet glosses
2013	SMT	750	Machine Translation evaluation
2014	headlines	750	newswire headlines
2014	OnWN	750	OntoNotes, WordNet glosses
2014	Deft-forum	450	forum posts
2014	Deft-news	300	news summary
2014	Images	750	image descriptions
2014	Tweet-news	750	tweet-news pairs
2015	Images	750	image descriptions
2015	headlines	750	newswire headlines
2015	answers-students	750	student answers
2015	answers-forum	375	forum answers
2015	belief	375	forum

original tree kernel when a vector composition function with specific ideal properties is used. The software is available here.[10]

5 DATASETS AND EXPERIMENT SETTINGS

The STS datasets (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015) are constructed from various sources associated with different domains, e.g newswire headlines, paraphrases, video description, image captions, machine translation evaluation, Twitter news and messages, forum data, glosses combination of OntoNotes, FrameNet and WordNet, etc. Only in STS 2012, the train and test datasets are provided, since STS 2013 onward, no new training dataset is given, but only the new test dataset, whereas datasets in previous years can be used for training. Except the setup in STS 2012 where several of test sets have designated training data, the STS 2013, 2014 setups are similar to STS 2015 with no domain-dependent training data. This domain-independent character of STS data is a great challenge for any system to achieve consistent performance. The detail of datasets described in Table 1.

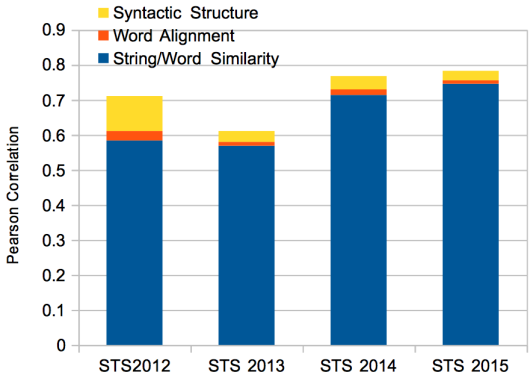

Figure 6: Component Analysis.

[10]https://code.google.com/p/distributed-tree-kernels

6 EVALUATIONS AND DISCUSSION

The results are obtained with Pearson correlation, which is the official measure used in both tasks.[11] The overall result is computed by the Weighted Mean of the Pearson correlations on individual datasets which is weighted according to the number of sentence pairs in that dataset. We compare our system's performance with the baseline and the top three systems in each STS competition in years 2012, 2013, 2014 and 2015.

Performance Comparison on all STS Datasets. Tables 2,3,4, and 5 show our system performance in each year. In overall, Table 6 shows the side-by-side comparison between our system and the baseline, the DKPro and the state-of-the-art (SOTA) systems on all STS datasets. This confirms our stable and consistent performance which always overcomes the baseline (large margin 20-27%) and DKPro (4-13%) , and achieves better or competitive results to SOTA systems.

Comparison to DKPro. Table 6 shows that though we adopt some string and word similarity features from DKPro, our system always outperforms DKPro. The main difference between our system and DKPro is that by adding two important modules of processing word alignment and syntactic structure, we consider more linguistic aspects in semantic inference leading to more robust and comprehensive capability to compute the semantic similarity. This proves that this approach of multi-layer infrastructure optimizes the system performance by delegating and capturing various linguistic phenomena by proper semantic layers, leading to higher precision and correlation.

Component Analysis. Figure 6 presents the analysis for each individual component in our STS system. It shows the significance of each layer into the overall performance on STS 2012, 2013, 2014 and 2015 datasets. Despite the fact that string and word similarity layer occupies a larger portion in the overall performance, the significance of other semantic layers is undenied. The design of multi-layer system im-

[11]http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

Table 2: Evaluation Results on STS 2012 datasets.

System	MSRpar	MSRvid	SMTeur	OnWN	SMTnews	Mean
Baseline	0.433	0.30	0.454	0.586	0.391	0.436
DKPro	0.62	0.808	0.376	0.657	0.462	0.584
UKP (1st)	0.683	0.874	0.528	0.664	0.494	0.677
Takelab (2nd)	0.734	0.880	0.477	0.680	0.399	0.675
SOFT-CARDINALITY (3rd)	0.641	0.856	0.515	0.711	0.483	0.671
ADW (Pilehvar et al., 2013)	0.694	0.887	0.555	0.706	0.604	**0.711**
OurSystem (OS)	**0.748**	**0.894**	0.458	**0.755**	0.505	**0.711**

Table 3: Evaluation Results on STS 2013 datasets.

System	FNWN	headlines	OnWN	SMT	Mean
Baseline	0.215	0.540	0.283	0.286	0.364
DKPro	0.385	0.706	0.784	0.317	0.569
UMBC_EBIQUITY_PairingWords (1st)	0.582	0.764	0.753	0.380	**0.618**
UMBC_EBIQUITY_galactus (2nd)	0.743	0.705	0.544	0.371	0.593
deft-baseline (3rd)	0.653	0.843	0.508	0.327	0.580
OurSystem (OS)	0.450	0.732	**0.843**	0.356	**0.611**

Table 4: Evaluation Results on STS 2014 datasets.

Systems	deft-forum	deft-news	headlines	images	OnWN	tweet-news	Mean
Baseline	0.353	0.596	0.510	0.513	0.406	0.654	0.507
DKPro	0.452	0.713	0.697	0.777	0.819	0.722	0.714
DLS@CU (1st)	0.483	0.766	0.765	0.821	0.859	0.764	0.761
MeerkatMafia (2nd)	0.471	0.763	0.760	0.801	0.875	0.779	0.761
NTNU (3rd)	0.531	0.781	0.784	0.834	0.850	0.676	0.755
OurSystem (OS)	0.508	0.762	0.765	0.818	**0.896**	0.749	**0.768**

Table 5: Evaluation Results on STS 2015 datasets.

System	ans-forums	ans-students	belief	headlines	images	Mean
Baseline	0.445	0.665	0.652	0.531	0.604	0.587
DKPro	0.696	0.712	0.699	0.766	0.808	0.746
DLS@CU-S1 (1st)	0.739	0.773	0.749	0.825	0.864	**0.802**
ExBThemis-themisexp (2nd)	0.695	0.778	0.748	0.825	0.853	0.794
DLS@CU-S2 (3rd)	0.724	0.757	0.722	0.825	0.863	0.792
OurSystem (OS)	0.713	0.744	0.733	0.808	0.858	**0.783**

Table 6: Comparison on all STS datasets.

Settings	2012	2013	2014	2015
Gain/Baseline	0.275	0.247	0.261	0.196
Gain/DKPro	0.127	0.042	0.054	0.037
Dist2SOTA	0.034	-0.007	0.007	-0.019

proves the overall performance from 3.7-12.7% more by better robustness and comprehension to handle more complicated semantic information via deeper semantic layers.

Accordingly, we can claim that our system consistently and stably performs at the state of the art or top-tier level on all STS datasets from 2012 to 2015. The framework of four different semantic layers helps our system handle heterogeneous data from STS successfully. By delegating and assigning different semantic layers which deal with different types of information, the system can cope with and adapt to any unknown domain data. This hypothesis is proven by the constant performance on various datasets derived from different domains in STS.

7 CONCLUSION AND FUTURE WORKS

In this paper, we investigated the variance of system performance in the STS task, then we presented a novel framework to solve the greatest challenge of

domain-independent data for Semantic Textual Similarity task. We unify the task into four main layers of processing to exploit the semantic similarity information from different presentation levels (lexical, string, syntactic, alignment) to overcome the variance of system's performance on data derived from various sources. Our framework is implemented and evaluated on all STS datasets and consistently achieves either state of the art or near state-of-the-art performance in regard to the top three best systems in every STS competition from 2012 to 2015.

REFERENCES

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., and Wiebe, J. (2015). SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, CO. Association for Computational Linguistics.

Agirre, E., Baneab, C., Cardiec, C., Cerd, D., Diabe, M., Gonzalez-Agirrea, A., Guof, W., Mihalceab, R., Rigaua, G., and Wiebeg, J. (2014). Semeval-2014 task 10: Multilingual semantic textual similarity. *SemEval 2014*, page 81.

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. Association for Computational Linguistics*. Citeseer.

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-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

Allison, L. and Dix, T. I. (1986). A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23(5):305–310.

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.

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-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 435–440. Association for Computational Linguistics.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL (1)*, pages 238–247.

Barrón-Cedeno, A., Rosso, P., Agirre, E., and Labaka, G. (2010). Plagiarism detection across distant language pairs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 37–45. Association for Computational Linguistics.

Berant, J., Dagan, I., and Goldberger, J. (2012). Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

Broder, A. Z. (1997). On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE.

Budanitsky, A. and Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.*, 32(1):13–47.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611.

Galitsky, B. (2013). Machine learning of syntactic parse trees for search and classification of text. *Engineering Applications of Artificial Intelligence*, 26(3):1072–1091.

Glickman, O. and Dagan, I. (2004). Acquiring lexical paraphrases from a single corpus. *Recent Advances in Natural Language Processing III. John Benjamins Publishing, Amsterdam, Netherlands*, pages 81–90.

Guo, W. and Diab, M. (2012). Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press.

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

Han, L., Martineau, J., Cheng, D., and Thomas, C. (2015). Samsung: Align-and-differentiate approach to semantic textual similarity. *SemEval-2015*, page 172.

Hänig, C., Remus, R., and De La Puente, X. (2015). Exb themis: Extensive feature extraction from word alignments for semantic textual similarity. *SemEval-2015*, page 264.

Harris, Z. S. (1968). *Mathematical structures of language*. Interscience Publishers.

Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

Lin, D. (1998). An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

Lyon, C., Malcolm, J., and Dickerson, B. (2001). Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 118–125.

Marsi, E., Moen, H., Bungum, L., Sizov, G., Gambäck, B., and Lynum, A. (2013). Ntnu-core: Combining strong features for semantic similarity. In *In* SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

Meadow, C. T. (1992). *Text information retrieval systems*. Academic Press, Inc.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Milne, D. and Witten, I. H. (2013). An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194:222–239.

Moschitti, A. (2006). Efficient convolution kernels for dependency and constituent syntactic trees. In *Machine Learning: ECML 2006*, pages 318–329. Springer.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

Pilehvar, M. T., Jurgens, D., and Navigli, R. (2013). Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *ACL (1)*, pages 1341–1351.

Plotkin, G. D. (1970). A note on inductive generalization. *Machine intelligence*, 5(1):153–163.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Sahlgren, M. (2006). The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.

Salton, G. and McGill, M. J. (1983). Introduction to modern information retrieval.

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-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 441–448. Association for Computational Linguistics.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, volume 12, pages 44–49. Manchester, UK.

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

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Sultan, M. A., Bethard, S., and Sumner, T. (2014a). Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230.

Sultan, M. A., Bethard, S., and Sumner, T. (2014b). Dls@ cu: Sentence similarity from word alignment. *SemEval 2014*, page 241.

Sultan, M. A., Bethard, S., and Sumner, T. (2015). Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 148–153.

Surdeanu, M., Ciaramita, M., and Zaragoza, H. (2011). Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2):351–383.

Turney, P. D., Pantel, P., et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Vo, N. P. A., Caselli, T., and Popescu, O. (2014). Fbk-tr: Applying svm with multiple linguistic features for cross-level semantic similarity. *SemEval 2014*, page 284.

Wise, M. J. (1993). String similarity via greedy string tiling and running karp-rabin matching. *Online Preprint, Dec*, 119.

Wise, M. J. (1996). Yap3: Improved detection of similarities in computer program and other texts. In *ACM SIGCSE Bulletin*, volume 28, pages 130–134. ACM.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Zanzotto, F. M. and Dell'Arciprete, L. (2012). Distributed tree kernels. *arXiv preprint arXiv:1206.4607*.