Journal of Theoretical and Applied Information Technology
30th November 2016. Vol.93. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COMPARISON OF DISTRIBUTIONAL SEMANTIC MODELS FOR RECOGNIZING TEXTUAL ENTAILMENT

[1]YUDI WIBISONO,[2]DWI HENDRATMO WIDYANTORO,[3]NUR ULFA MAULIDEVI
[1]School of Electrical Engineering and Informatics, Institut Teknologi Bandung
E-mail: [1]yudi,wibisono@students.itb.ac.id, [2]dwi@stei.itb.ac.id, [3]ulfa@stei.itb.ac.id

ABSTRACT

Recognizing Textual Entailment (RTE) is an important task in many natural language processing. In this paper we investigate the effectiveness of distributional semantic model (DSM) in RTE task. Word2Vec and GloVe are recent methods that suitable for learning DSM using a large corpus and vocabulary. Seven distributional semantic models (DSM) generated using Word2Vec and GloVe were compared to get the best performer for RTE. To our knowledge, this paper is the first study of various DSM on RTE. We found that DSM improves entailment accuracy, with the best DSM is GloVe trained with 42 billion tokens taken from Common Crawl corpus. We also found the size of vocabulary size in DSM does not guarantee higher accuracy.

Keywords: *Recognizing Textual Entailment, Distributional Semantic Model, Text Alignment*

1. INTRODUCTION

Recognizing Textual Entailment (RTE) is the task of detecting directional semantic entailment relation between a text pairs <text, hypothesis>. Text T entails hypothesis H, if H can be inferred from T using common knowledge. RTE task is important because many natural language processing (NLP) problems, such as information extraction, relation extraction, summarization or machine translation, rely on it [1]. As example, Figure 1 shows a pair of text *T* and hypothesis *H*. Although words and structures of text and hypothesis are different, *H* is entailed from *T*.

T: Every year Israel jails individuals simply because they refuse to perform military service for reasons of conscience.

H: People are willing to risk imprisonment rather than perform military service.

Entail: TRUE

Figure 1. Example from RTE3 dataset

RTE is related with semantic textual similarity (STS). The difference is that textual entailment is one directional [2], and STS is bidirectional. STS compared two sentences that are relatively equal in length, while the textual entailment compared hypothesis that are shorter than text.

Common approach for RTE is using textual similarity, that finds similarity between two texts by using simple lexical matching method [3]. Similarity score returns number of lexical units that occur in both input texts.

The main problem is there are so many word variation in RTE [4]. For example, there is semantic similarity between sentence "I own a cat" and "I have an animal", but standard lexical similarity will fail to identify similarity between these texts [3]. Identifying word similarity of texts may improve RTE system.

For handling word variation, some systems employed external semantic resources e.g. WordNet, DIRT, Wikipedia, VerbNet and Framenet [7-8]. System with external semantic resources has better accuracy than without external resources [9]. Unfortunately, manually build database like WordNet and VerbNet has limited vocabulary coverage. In order to overcome this weakness, DSM has been proposed to build automatically semantic resources based on unlabeled corpus. We use Word2Vec [11] and GloVe [12] that designed to generate DSM for large corpus.

There are few papers that employed large DSM in RTE. Bjerva [19] used Word2Vec trained with 1 billion tokens from Wikipedia, Teranaka [20] used Word2Vec trained with Japan Wikipedia corpus. They employed vector distance as a feature for machine learning, but the effect of DSM is not explained. Zhao [21] used Word2Vec trained using Google News Corpus and concluded that DSM has no effect for RTE accuracy. To our knowledge, our

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

paper is the first study of various distributional semantic model on RTE and the first that use GloVe method.

This paper compares some distributional semantic models (DSM) for RTE. In our approach, textual alignment is developed with semantic distributional word similarity model to determine entailment relation between text and its hypothesis. If the hypothesis is semantically equivalent with text (entailed) then their words or expression should be aligned. Entailment relation is predicted by the alignment quality. Align and penalize algorithm in [6] is adapted and semantic distributional model replace LSA (Latent Semantic Analysis).

This paper shows that entailment accuracy is improved by DSM, especially by GloVe with 42 billion tokens. We also found the size of vocabulary size in DSM does not guarantee higher accuracy.

The rest of the paper is organized as follows. In section 2, related works in this area are presented. Section 3 describes distributional semantic model, and section 4 describes our method. In section 5, experiments are described including discussion of results. Concluding remarks are presented in section 6.

2. RELATED WORK

Lack of coverage is the first problem in semantic resources of textual entailment system. Semantic resource, e.g. WordNet that is built manually by linguists, has limited number of words, and has disadvantage in adding new words. The second problem is non-parametric database, so the size of vocabulary is linear with the size of corpus.

We adapted UMBC-Ubiquity-Core model [6]. This system is the best system for SEM 2013 semantic textual similarity task by using simple align and penalize algorithm. Alignment model in textual entailment defines correspondence between the words of the hypothesis and the words of text. Penalty is given when the correspondence words is not found.

Alignment model assumes that if there are two semantically similar sentences, its words or phrases can be aligned. Alignment quality will be used as similarity measurement. For handling vocabulary variation, UMBC using a combination of Latent Semantic Analysis (LSA) and WordNet to measure similarity between two words.

3. DISTRIBUTIONAL SEMANTIC MODEL

Distributional Semantic Model (DSM) assumes that semantically similar words were likely to co-occur within the same context or same word group. DSM represents word context by using vector, and word similarity is calculated by geometric distance of two vectors. This semantic vector is used as features in many applications.

DSM has been employed in NLP tasks, i.e. semantic relatedness [10], synonym detection [11], concept categorization [12], selection preferences [13], and analogy [14]. In semantic relatedness especially in distinguishing similarity from relatedness, DSM had better performance than performances of state of the art techniques [10]. There are some popular DSM i.e. Word2Vec [11], GloVe [12], and Paragram [5]. These methods have advantages in processing very large unannotated corpus, and providing pre-trained model and ready-made tools.

3.1 Word2Vec

Word2Vec employs neural network to return semantic vector model from corpus. It has two architectures i.e. skip-gram and continuous bag-of-words (CBOW). Skip-gram predicts contexts as neighboring words in particular range when accepting an input word. CBOW maximizes conditional probabilities of word occurrence when accepting contexts (i.e. neighboring words). [11]

In analogy task, Mikolov et al. [13] introduced new evaluation scheme that employs structure of word vector space by using some dimensions of difference. As an example, analogy “king is to queen as man is to woman” can be represented as vector space in “king − queen = man − woman”.

Baroni et al. [10] compared Word2Vec and count models. The best performing vectors are 400-dimensional, 5-word context window, with 10 negative samples and subsampling. All models consist of about 2.8 billion tokens constructed by concatenating UK WaC, English Wikipedia, and British National Corpus [10].

3.2 Global Vector

Global Vector (GloVe) uses statistical counting global word co-occurrence non-zero matrix. There is a claim that global counting usage made Glove better than Word2Vec for analogy task, semantic similarity, and named-entity recognition, but this claim is disputed by Levy [14]. Glove employs ratio of co-occurrence probabilities to eliminate non-discriminative words [12].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.3 Paragram

Paragram is parametric paraphrase model built using skip-gram Word2Vec [11] and recursive neural network GloVe [12]. Wieting [5] proposed this model to improve weaknesses of Paraphrase Database (PPDB). PPDB is semantic resources containing database of 220 million paraphrase pairs [15].

Paragram is developed because there are three weaknesses of PPDB [5]. The first weakness is incomplete coverage since both phrases are assumed co-exist in the database. Non parametric is the second weakness of PPDB, the number of phrase pairs increase in line with dataset used to construct the database. Large size of phrase pairs will complicate database usage. Last, low quality of confidence estimation by PPDB that is heuristically feature combination. Paragram is built by using PPDB corpus, and then tuned on by Simlex-99 and WordSim-353.

4. DSM+ALIGN ENTAILMENT

Figure 2 shows our proposed system architecture. For alignment, we apply modified align-and-penalize [6]. We add named entity alignment, and replaced LSA and WordNet with distributed semantic model (DSM).

Preprocessing consists of tokenization, stop word removal, part of speech (POS) tagging, and named-entity recognition (NER) using Stanford coreNLP library [16]. Each token is categorized into three POS tags, i.e. verb, noun, and other word. NER returns four types of named-entities, namely number, location, currency, and date.

Figure 3 shows isEntail algorithm, that apply align and penalize algorithm for entailment. It begins by matching each list of named-entities (see Figure 4). The next step is matching verb and noun, then matching other words (see Figure 5). If a word or phrase from *H* has no pair in *T*, penalty is applied. Final score is calculated from score and penalty returned by *namedEntityAlign* and score and penalty returned by wordAlign. Unlike UMBC that has bidirectional alignment score, alignment score of textual entailment is calculated in one directional. Entailment is true if final score is above predefined threshold.

Threshold *t* in *isEntail* algorithm in Figure 3 is the best split value of entailment label classification. We employ gain ratio of C4.5 to find the best threshold using score attribute of training data. In C4.5 application, our aim is not the best attribute selection because training data consists of only one attribute (i.e. score) and one label (i.e. entailment). Gain ratio is employed to separate score of training instances according to target classification, i.e. entailment label [17].

We investigate two pre-trained DSM i.e. Word2Vec [10-11] and Glove [12] using in align and penalize algorithm. Pre-trained Word2Vec and Glove have many models, based on different corpus in model building, as shown by Table 1. We employ two pre-trained Word2Vec and five pre-trained Glove.

Table 1. DSM description

DSM	Corpus	Vocab. Size	Dimension
Word2Vec_ Mikolov	Google News	3 million	300
Word2Vec_Baroni	UKWaC, Wikipedia, and British National Corpus	2.8 billion	400
Glove_sl999	PPDB tuned on SimLex999	N/A	300
Glove_ws353	PPDB tuned on WordSim353	N/A	300
Glove_6B	Wikipedia 2014 + GigaWord 5	N/A	300
Glove_42B	Common Crawl 42 billion tokens	1.9 million (uncased)	300
Glove_840B	Common Crawl 840 billion	2.2 million (cased)	300

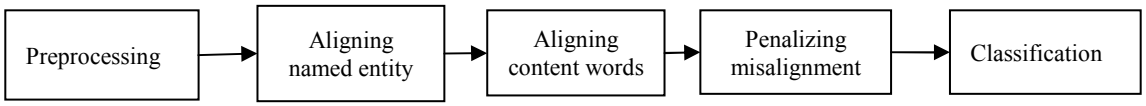

Figure 2. System overview

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 1: *isEntail*(*H*, *T*, *t*)
Description: Predicting whether H entail T using align and penalize.
Input:
H, T: text pair to be classified
t: threshold constant for classification
Output: *label*: boolean, true if *H* entails *T*

Algorithm
1. *tokenH*← *tokenize* (*H*)
2. *Hne* ← {(*term*): *term*∈*tokenH*∧*NER*(*term*) ∈ {number, location, date, currency}}
3. *tokenT*← *tokenize* (*T*)
4. *Tne* ←{(*term*): *term*∈*tokenT*∧*NER*(*term*) ∈ {number, location, date, currency}}
5. (*scoreNEAlign*, *penaltyNEAlign*) ← *namedEntityAlign*(*Tne*, *Hne*)
6. (*scoreWordAlign*, *penaltyWordAlign*) ← *wordAlign*(*tokenH*,*tokenT*)
7. $score \leftarrow \frac{scoreNEAlign+scoreWordAlign}{|H|} - \frac{penaltyNEAlign+penaltyWordAlign}{|H|}$
8. **if** score > t **then** label ← true
else label ←false

Figure 3 isEntail algorithm

Figure 4 shows details of *namedEntityAlign* algorithm. Alignment begins by processing each named-entity in *H*. If a named-entity has a pair with named-entity in *H*, score is increased. If there is no pair, total penalty is increased. This process is applied to all named-entities in *H*.

Figure 5 shows details of *wordAlign* algorithm. We use three kinds of POS tag, i.e. verb, noun, and other. Like *namedEntityAlign* algorithm, this algorithm finds word pairs by matching words in H and words in T. The difference is that if there are no exact matches, semantic relatedness is identified by function isSemanticRelated between token *t1* and token *t2*. Given *t* as threshold, the function *isSemanticRelated (t1, t2)* is calculated using following formula. DMS is then applied in the function similarity.

$$isSemanticRelated \leftarrow \max_{t2 \in tokens} similarity(t1, t2) > t$$

Table 2 shows the best penalty constants for each named entity or POS tags. We used grid search on training data to find optimal penalty constants.

Table 2. Penalty constants

Named Entity or POS tag	Penalty value
Number	1
Location	0.5
Date	0.5
Currency	0.5
Verb & noun	0.5
Other word	0.5

5. EXPERIMENT

Our experiments aim to investigate the best DSM for RTE. We use RTE3 dataset [18], consisting of 800 text-hypothesis pairs in an 800/800 train/test split. We use seven DSM for word similarity (see Table 1). Entailment threshold is determined by using gain ratio of alignment score of training data.

Testing accuracy is shown in Table 3. The best accuracy is achieved by Glove_42B with accuracy of 67.75%. We found that Glove_ws353, Glove_sl999 and Word2Vec_Baroni have lower performances than generic model, i.e. Glove_42B although they were designed specifically for paraphrase that is more related with entailment task.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 2: *namedEntityAlign* (*Hne*, *Tne*)
Description: Align and penalize based on named entity from H and T
Input:
1. *Hne*: list of token from H with particular named entity
2. *Tne*: list of token from T with particular named entity
3. *locationPenalty*: real
4. *currencyPenalty*: real
5. *numberPenalty*: real
6. *datePenalty*: real

Output:
1. *score*: alignment score for name entity in H and T
2. *penalty*: alignment penalty for name entity in H and T

Algorithm

1. *score* ← 0
2. *locationH*← {(*term*): *term* ∈ *Hne* ⋀ *NamedEntity* (*term*)=location}
3. *locationT* ←{(*term*): *term* ∈ *Tne* ⋀ *NamedEntity* (*term*)=location}
4. **foreach** (*locH*:*locationH*)
5. **if** (*locH*∈*locationT*) **then** *score* ← *score*+1 **else** *penalty*← *penalty*+*locationPenalty*

6. *currencyH*←{(*term*): *term* ∈ *Hne* ⋀ *NamedEntity* (*term*)=currency }
7. *currencyT*←{(*term*): *term* ∈ *Tne* ⋀ *NamedEntity* (*term*)=currency }
8. **foreach** (*currH*:*currencyH*)
9. **if** (*currH*∈*currencyT*) **then** *score* ← *score*+1 **else** *penalty*← *penalty*+*currencyPenalty*

10. *numberH*←{(*term*): *term* ∈ *Hne* ⋀ *NamedEntity (term)*=number }
11. *numberT*←{(*term*): *term* ∈ *Tne* ⋀ *NamedEntity (term)*=number }
12. **foreach** (*numH*:*numberH*)
13. **if** (*numH*∈*numberT*) **then** *score* ← *score* +1 **else** *penalty*← *penalty* +*numberPenalty*

14. *dateH*←{(*term*): *term* ∈ *Hne* ⋀ *NamedEntity* (*term*)=date}
15. *dateT*←{(*term*): *term* ∈ *Tne* ⋀ *NamedEntity* (*term*)=date}
16. **foreach** (*dH*:*dateH*)
17. **if** (*dH*∈*dateT*) **then** *score* ← *score* +1 **else** *penalty*← *penalty* +*datePenalty*

Figure 4. namedEntityAlign algorithm

Table 3. RTE testing accuracy using various DSM, ordered ascending

Model	Glove_6B	Word2Vec_Mikolov	Word2Vec_Baroni	Glove_ws353	Glove_840B	Glove_sl999	Glove_42B
Accuracy (%)	64.75	65.75	66.00	66.50	66.88	67.38	**67.75**

We also found that the size of vocabulary size in DSM does not guarantee a higher accuracy. Although performance of Glove_42B that has 42 billion tokens is better than performance of Glove_6B (6 billions token), performance of Glove_840B (840 billions token) is lower than Glove_42B.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 3: *wordAlign*(*tokenH*,*tokenT*)
Description: Align and penalize based on term from token H and token T
Input:

1. *tokenH*: list of token from H
2. *tokenT*: list of token from T

Output:

1. *score*: alignment score for name entity in H and T
2. *penalty*: alignment penalty for name entity in H and T

Algorithm

1. *Hv*← {(*term*): *term* ∈ *tokenH* ⋀ *POS* (*term*)=verb}
2. *Tv*← {(*term*): *term* ∈ *tokenT* ⋀ *POS* (*term*)=verb}
3. **Foreach**(*v*: *Hv*)
4. **if** (*v*∈ *Tv*∨ (*isSemanticRelated* (*v*,*Tv*)) **then** *score* ← *score* + *SemanticRelatedValue*(*v*,*Tv*)
   else *penalty*← *penalty*+*verbPenalty*
5. *Hn*← {(*term*): *term* ∈ *tokenH* ⋀ *POS* (*term*)=noun}
6. *Tn*← {(*term*): *term* ∈ *tokenT* ⋀ *POS* (*term*)=noun}
7. **Foreach**(*n*: *Hn*)
8. **if** (*n*∈ *Tn*∨ (*isSemanticRelated* (*n*,*Tn*)) **then** *score* ← *score* + *SemanticRelatedValue*(*n*,*Tn*)
   else *penalty*← *penalty*+*nounPenalty*
9. Ho ← {(*term*): *term* ∈ *tokenH* ⋀ *POS*(*term*) ≠ verb ⋀ *POS*(*term*) ≠ noun }
10. To ← {(*term*): *term* ∈ *tokenT* ⋀ *POS*(*term*) ≠ verb ⋀ *POS*(*term*) ≠ noun }
11. **Foreach**(*o*: *Ho*)
12. **if** (*o*∈ *To*∨ (*isSemanticRelated*(*o*,*To*)) **then** *score* ← *score* + *SemanticRelatedValue*(*o*,*To*)
    else *penalty*← *penalty*+*verbPenalty*

Figure 5. wordAlign algorithm

The main issue in word semantic similarity used in textual entailment is how to distinguish between similarity and relatedness. Similarity contains closer taxonomic relation, such as synonym whereas the second is broader and topical relation. For example, cups and coffee often get a high similarity score in some external lexical database model because it is often appeared in same context. We know cups and coffee is not similar, and cups is more similar with word mug which have same function and shape [5].

The best model should be able to distinguish between similar and related words. As an example shown by Figure 6, Although "dagbladet" and "osloposten" is related because both are names of Norwegian newspapers, it is not semantically similar. Glove_42B returns low semantic similarity score for the pair, which is good, but another models return high score for this pair.

T: Big article in the Norwegian Newspaper called **Osloposten** about the Scientology company U-MAN. Interviews with Magne Berge and the author of Operation Clambake.

H: **Dagbladet** is a Norwegian newspaper.

Entailment: FALSE

Figure 6. Example 1: Glove_42B is better than others

Another example is shown by Figure 7. Glove_42B is accurate by found no word pair for "resided". On the other hand, Word2Vec model found word pair "resided" and "evacuated", and Glove_ws353 and Glove_sl999 found word pair "resided" and "homes". It is shown that Glove_42B can distinguish between related and similar words.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

T: After arresting Omar, police **evacuated** up to 100 **homes** and sent a bomb squad into the Small Heath neighborhood (search) in Birmingham, a city some 120 miles northwest of London.

H: Omar **resided** in the Small Heath neighborhood.

Entailment: FALSE

Figure 7. Example 2: Glove_42B is better than others

There are still problems in using DSM. As shown by Figure 8, all models return highest score for "defeating" and "defeated". However, if these two words are exchanged, subject and object can be misidentified.

T: Sandra Goudie was first **elected** to Parliament in the 2002 elections, narrowly winning the seat of Coromandel by **defeating** Labor candidate Max Purnell and pushing incumbent Green MP Jeanette Fitzsimons into third place.

H: Sandra Goudie was **defeated** by Max Purnell.

Entailment: FALSE

Figure 8. Example 3: false positive case

Another problem is coverage. In Figure 9, all DSMs fail to identify semantic relatedness between "smoking pot" and "drugs".

T: It didn't happen because the cream of England's thugs was **smoking pot** which is easily and legally available in the Netherlands.

H: **Drugs** in Holland are easily bought.

Entailment: TRUE

Figure 9. Example 4: false negative (low coverage of noun)

All models also fail to match "agreed to prepare "and "will be" in example shown by Figure 10. Both examples show low coverage problem of DSM.

T: Czech and Slovak leaders announced early today that they had **agreed to prepare** the splitting up of Czechoslovakia into two separate states.

H: Czech and Slovak **will be** split into two separate states.

Entailment: FALSE

Figure 10. Example 5: false negative (low coverage of verb)

6. CONCLUSION

In this paper, seven distributional semantic models have been investigated for recognizing textual entailment task. To our knowledge, this is the first study of various distributional semantic model on RTE.

Glove_42B model that employs GloVe method achieved the best accuracy of 67.75%. This model is trained using Common Crawl corpus with 42 billion tokens and 1.9 million vocabularies.

For future research, we will explore and build custom DSM that is able to distinguish similar and relatedness words more accurately. Since our findings are specific on RTE3 dataset, we will apply the proposed method for another dataset.

REFERENCES:

[1] Dagan, Ido, Oren Glickman, and Bernardo Magnini. "The PASCAL recognising textual entailment challenge." Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising textual entailment. Springer Berlin Heidelberg, 2006. 177-190.

[2] Ion Androutsopoulos, Prodromos Malakasiotis. "A Survey of Paraphrasing and Textual Entailment Methods." Journal of Artificial Intelligence Research 38 (2010) 135-187

[3] Rada Mihalcea, Courtney Corley, Carlo Strapparava. "Corpus-based and Knowledge-based Measures of Text Semantic Similarity", American Association for Artificial Intelligence, 2006.

[4] Yudi Wibisono, Dwi H. Widyantoro, Nur Ulfa Maulidevi. "Sentence Extraction in Recognition Textual Entailment Task.". Proceedings of International Conference on Data and Software Engineering, 2014.

[5] Wieting, John, Mohit Bansal, Kevin Gimpel, and Karen Livescu. "Towards universal paraphrastic sentence embeddings." arXiv preprint arXiv:1511.08198 (2015).

[6] Han, Lushan, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. "UMBC EBIQUITY-CORE: Semantic textual similarity systems." InProceedings of the Second Joint Conference on Lexical and Computational Semantics, vol. 1, pp. 44-52. 2013.

[7] Adrian Iftene, Alexandra Balahur-Dobrescu. "Hypothesis transformation and semantic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

variability rules used in recognizing textual entailment." Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Association for Computational Linguistics, 2007.

[8] Marta Tatu, Dan Moldovan. "Cogex at RTE3." Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Association for Computational Linguistics, 2007.

[9] Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan. "The third pascal recognizing textual entailment challenge." In Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing, pp. 1-9. Association for Computational Linguistics, 2007.

[10] Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors." ACL (1). 2014.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In NIPS, pages 3111–3119.

[12] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.

[13] Tomas Mikolov, Wen-Tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In HLT-NAACL, pages 746– 751

[14] Levy, Omer, Yoav Goldberg, and Ido Dagan. "Improving distributional similarity with lessons learned from word embeddings." Transactions of the Association for Computational Linguistics 3 (2015): 211-225.

[15] Ganitkevitch, Juri, Benjamin Van Durme, and Chris Callison-Burch. "PPDB: The Paraphrase Database." HLT-NAACL. 2013.

[16] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

[17] Tom M. Mitchell, "Machine Learning," New York: McGraw-Hill, Inc., 1997.

[18] Giampiccolo, Danilo, Bernardo Magnini, Ido Dagan, and Bill Dolan. "The third pascal recognizing textual entailment challenge." In Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing, pp. 1-9. Association for Computational Linguistics, 2007.

[19] Bjerva, Johannes, Johan Bos, Rob Van der Goot, and Malvina Nissim. "The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity." Proceedings of SemEval (2014).

[20] Teranaka, Genki, Masahiko Sunohara, and Hiroaki Saito. "NAK Team's System for Recognition Textual Entailment at the NTCIR-11 RITE-VAL task."NTCIR-11 RITE-VAL (2014).

[21] Zhao, Jiang, Man Lan, Zheng-Yu Niu, and Yue Lu. "Integrating word embeddings and traditional NLP features to measure textual entailment and semantic relatedness of sentence pairs." In 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, 2015.