# Sentential Paraphrase Generation for Agglutinative Languages Using SVM with a String Kernel

**Hancheol Park**[1], **Gahgene Gweon**[1], **Ho-Jin Choi**[2], **Jeong Heo**[3], **Pum-Mo Ryu**[3]
[1]Department of Knowledge Service Engineering, KAIST
[2]Department of Computer Science, KAIST
[3]Knowledge Mining Research Team, ETRI
{hancheol.park, ggweon, hojinc}@kaist.ac.kr
{jeonghur, pmryu}@etri.re.kr

## Abstract

Paraphrase generation is widely used for various natural language processing (NLP) applications such as question answering, multi-document summarization, and machine translation. In this study, we identify the problems occurring in the process of applying existing probabilistic model-based methods to agglutinative languages, and provide solutions by reflecting the inherent characteristics of agglutinative languages. More specifically, we propose and evaluate a sentential paraphrase generation (SPG) method for the Korean language using Support Vector Machines (SVM) with a string kernel. The quality of generated paraphrases is evaluated using three criteria: (1) meaning preservation, (2) grammaticality, and (3) equivalence. Our experiment shows that the proposed method outperformed a probabilistic model-based method by 12%, 16%, and 17%, respectively, with respect to the three criteria.

## 1 Introduction

Paraphrase generation (PG) is a useful technique in various natural language processing (NLP) applications, where it expands natural language expressions. In question answering systems, PG can be utilized to generate semantically equivalent questions. It can solve word mismatch problems when searching for answers (Lin and Pantel, 2001; Riezler et al., 2007). For multi-document summarization, it also helps to generate a summary sentence by identifying repeated information among semantically similar sentences (McKeown et al., 2002). In addition, for machine translation, paraphrasing can mitigate the scarcity of training data by expanding the reference translations (Callison-Burch, 2006).

In this study, we focus on a paraphrase generation approach, namely, sentential paraphrase generation (SPG), which takes a whole sentence as an input and generates a paraphrased output sentence that has the same meaning. Figure 1 shows an overview of the SPG process in general.

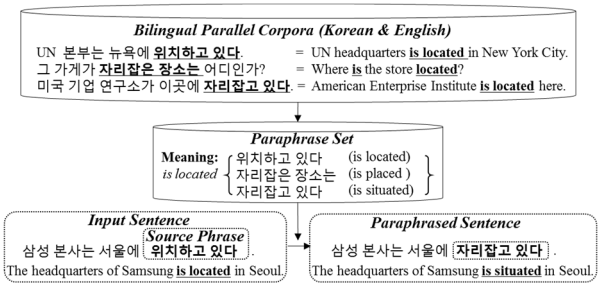


Figure 1: Example of the SPG process.

For example, let us assume that we would like to generate a paraphrased sentence using bilingual parallel corpora for a Korean input sentence "삼성본사는 서울에 위치하고 있다 (The headquarters of Samsung is located in Seoul)." For simplicity, in the examples used in this paper, we assume that an input sentence has only one source phrase to be substituted/ paraphrased. In our sample sentence, the source phrase is "위치하고 있다 (is located)." Currently, popular methods for SPG use phrase-based statistical machine


28th Pacific Asia Conference on Language, Information and Computation pages 650–657

translation (PBSMT) techniques (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Zhao et al., 2009; Wubben et al., 2010) with phrase-based paraphrase sets extracted from bilingual or monolingual parallel corpora. Such methods based on PBSMT use probabilistic-based models (e.g., a paraphrase model (PM) and a language model (LM)) to select the best phrase for substitution from a paraphrase set, which contains phrases that share the same meaning, to produce a paraphrased sentence. Probabilistic-based methods improve the system as the size of the corpora increases with increased frequency of the phrases. However, these methods tend to encounter two problems when applied to agglutinative languages (e.g., Korean, Japanese, and Turkish), which are morphologically rich languages.

The first problem is that it is very difficult to obtain a reliable probability distribution in agglutinative languages. Isolating (e.g., Chinese) and inflectional (e.g., Latin and German) languages employ fewer lexical variants to represent diverse grammatical functions or categories, whereas in agglutinative languages this process leads to an enormous number of possible inflected variants of a word. This is because a word is formed by combining at least one root, which represents a meaning, with various function or bound morphemes (e.g., postpositional particles and affixes). Furthermore, agglutinative languages suffer from the problem of resource scarcity (Wang et al., 2013). This problem becomes even more severe when obtaining an appropriate probability distribution for each variant, because the frequency of each phrase in a paraphrase set is less than in other languages, given the same quantity of corpora. In this study, therefore, we propose to use Support Vector Machines (SVM) for classification, which select the best paraphrase without employing probability information.

The second problem in using previously proposed probabilistic-based methods with agglutinative languages is that these methods lead to lower grammaticality because these methods do not consider the internal structure of a source phrase and the internal structures of dependent words of the source phrase. These methods take into account only the surface form distribution. It is very difficult to identify grammatically correct candidates in the paraphrase sets. This problem appears to be much more severe in agglutinative languages than in isolating or inflectional languages.

For this reason, in this study, we propose to utilize the similarity of syntactic categories, grammatical categories, and contextual information between the source phrase and its candidate paraphrases, when selecting the best paraphrase.

In this paper, we propose a novel SPG method that deals with the two problems mentioned above, for the Korean language, which is an agglutinative language. In the remainder of this paper, we review background literatures for our method on paraphrase generation in section 2; describe our proposed method in section 3, explain the experimental settings and results in section 4, and conclude in section 5.

## 2 Background

### 2.1 Probabilistic Model-Based Paraphrase

An SPG process begins with paraphrase phrases extraction from monolingual or bilingual parallel corpora. In this section, we review a popular paraphrasing method introduced by Bannard and Callison-Burch (2005). Since this is one of the very first studies to be conducted using bilingual parallel corpora and is a fundamental method in research on paraphrasing with bilingual parallel corpora, we used it as the baseline for our comparative experiment in this paper.

The method assumes that phrases that share commonly aligned foreign phrases are likely to be paraphrases of each other. For example, English phrases $e_1$ and $e_2$ that share commonly aligned foreign phrases $f$ can be regarded as paraphrases of each other and their "*paraphrase probability*" is expressed as follows:

$$p(e_2|e_1) = \sum_f p(e_2|f)\, p(f|e_1)$$

Given a source phrase $e_1$ in a new input sentence, the best paraphrase $\widehat{e_2}$ is chosen from candidate phrases $e_2$ as expressed in equation below:

$$\widehat{e_2} = argmax_{e_2:\, e_2 \neq e_1}\, p(e_2|e_1)$$
$$p(w_{-2}\, w_{-1}\, e_2\, w_{+1}\, w_{+2})$$

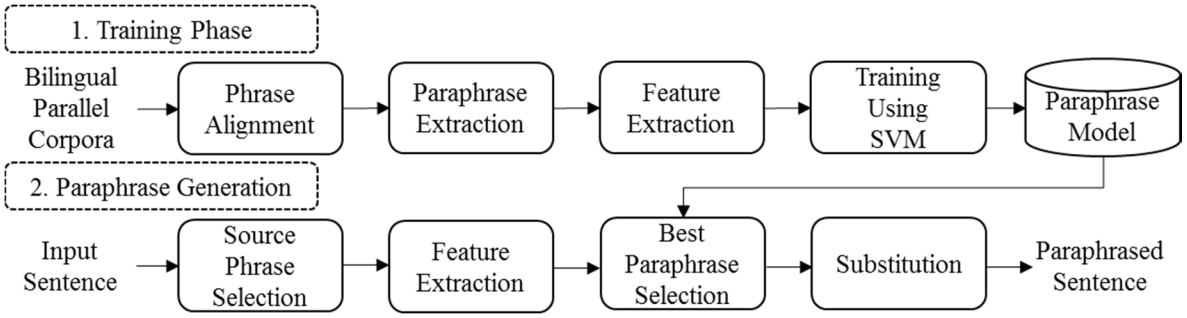


Figure 2: Overview of the proposed SPG method.

Since we use a trigram LM to decide how acceptable each $e_2$ is for a given input, $w_{-2}$ and $w_{-1}$ are the two words preceding $e_2$, and $w_{+1}$ and $w_{+2}$ are the two words following $e_2$. The phrase $e_1$ is substituted with the best paraphrase $\hat{e_2}$, which has the highest probability.

### 2.2 Classification Using SVM with a String Kernel

In this study, we propose an SPG method using SVM, instead of using the probabilistic-based model, which is used in the approach described in section 2.1.

An SVM is a linear classifier that finds a linear hyper-plane that separates positive and negative instances of labeled samples with the largest margin. This classifier is designed to reduce the generalization error rate, which is the ratio of incorrectly predicted classes to the novel inputs, because it is less overfitted to the training data set than other methods (Kozareva and Montoyo, 2006). With reference to sparseness of each lexical variant in agglutinative languages, it is also more tolerant than probabilistic-based models because it does not largely depend on the frequency of instances.

For problems that are not linearly separable, SVM uses a kernel function that implicitly transforms a non-linear problem into a higher-dimension space and makes the problem into a linearly separable one. A kernel is a similarity function between a pair of instances. In particular, since string kernels are useful in terms of measuring the similarity of non-fixed size feature vectors (e.g., text documents, the dependency tree of a sentence, and syntax trees) (Erkan et al., 2007), we used a string kernel given that the features that consider the morphological structures of words are variable in length. More specifically, we use the edit distance kernel function (Erkan et al., 2007) as follows:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \text{ edit\_distance}(\boldsymbol{x}_i, \boldsymbol{x}_j))$$

Here, *edit_distance* is defined as the Levenshtein distance between string $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. i.e., the minimum number of edits (deletions, insertions, or substitutions at the word level) required to transform one string into another. One of the advantages of using this kernel is that it takes into account the order of the strings in the structured data (e.g., a dependency path tree) as opposed to other string kernels (e.g., cosine similarity kernel) which consider only the common terms when measuring similarity.

In our study, the class for SVM is each phrase in a paraphrase set, which contains a source phrase. In addition, the SPG used in our study can be regarded as a multiclass classification problem. Thus, to solve this problem with a binary classifier, we adopted a "one-against-one" approach (Chang and Lin, 2011). This approach constructs $k(k-1)/2$ classifiers, where $k$ is the number of classes, with a training data set from two classes. A new data point is allocated to the class with the most votes during each binary classification.

## 3 Paraphrase Generation Using SVM with a String Kernel

This section describes our proposed SPG method, which uses an SVM with a string kernel. Figure 2 shows an overview of this method.

### 3.1 Training Phase

In the training phase, phrase alignment is first conducted manually using training sentences, which is composed of bilingual parallel corpora of

the Korean and English languages, as shown in Figure 1. Phrase alignment can be automatically conducted using the GIZA++ toolkit (Och and Ney, 2003) and phrase alignment heuristics (Koehn et al., 2003). However, in this study, we evaluate the performance of our generation method independently from the quality of automatic word or phrase alignment algorithms. Therefore, we conduct manual alignment, which is much more accurate than automatic alignment. In addition, applying these alignment tools to the Korean language is not appropriate because they only obtain a few correct results.

We align Korean phrases that range from unigrams to trigrams with English. For example, "뉴욕에 위치하고 있다 = is located in New York City" in the bilingual parallel corpora shown in Figure 1 can be aligned as follows:

- 뉴욕에 (unigram)

  = in New York City
- 뉴욕에 위치하고 (bigram)

  = located in New York City
- 뉴욕에 위치하고 있다 (trigram)

  = is located in New York City

With these aligned phrases, we extract paraphrase sets by grouping phrases that have common foreign phrases (i.e., English) because they are likely to have the same meaning (e.g., is located = 위치하고 있다, 자리잡은 장소는, 자리잡고 있다 in Figure 1).

Next, for feature extraction, three types of features are generated for the training phrases in paraphrase sets, referring to the sentences that the phrases are originally contained in: syntactic categories (SC), grammatical categories (GC), and contextual information (CI). For each type of feature, characteristics of the training phrase as well as the dependent words that precede and follow the training phrase in a Korean training sentence are extracted.

These three features help enhance the paraphrasing method using the agglutinative languages. Using the SC and GC features helps maintain the grammaticality of the source phrase. However, given the high variation in postpositional particles or affixes in agglutinative languages, there is a low probability of matching the SC and GC features in the source and training phrases. Therefore, by considering the SC and GC features of the dependent words in addition to the features of the source phrase, our method considers the context of the source phrase in terms of grammaticality to find the best candidate for paraphrasing. The CI features have a similar purpose in that they consider the context of the word sense of neighboring words.

- **Syntactic Categories (SC):** This feature helps to select a phrase with an acceptable syntactic type based on the structure of a given sentence. Morphological analysis is conducted for three phrases: the training phrase as well as the two dependent words preceding and following the training phrase. Based on the result of the morphological analysis, features are extracted such as phrase type (e.g., noun phrase (NP), verb phrase (VP)), case (e.g., subject (SBJ), and object (OBJ)), and tags of morphemes (e.g., pronoun (np) and case particle (jc)) for the three phrases.
- **Grammatical Categories (GC):** This feature helps to select a phrase that preserves the grammatical categories of a source phrase. Grammatical categories are extracted for the training phrase as well as the dependent words preceding and following the training phrase. They are extracted by considering the affixes of each phrase or word. Sample features for GC include the sentence type (e.g., interrogative sentence (INT), declarative sentence (DEC)), voice (e.g., passive (PAS), active (ACT)), and tense (past (PAST), present (PRES), and future (FUTU)). This feature is labeled as "N/A" if a corresponding feature does not exist.
- **Contextual Information (CI):** This feature helps to select a phrase that has the same word sense as a source phrase. Contextual information is extracted by taking the roots for the preceding and following dependent words.

The features are represented as a string instead of a numerical feature vector since a string kernel is used. Finally, phrases in a paraphrase set with identical meanings and corresponding features are

| Method | Sentences |
|---|---|
| TS | **[이것은]**[1] **[무엇으로]**[2] **[이용되는가]**[3]? (What is this utilized for?) |
| Baseline | **[그것은]**[1] **[어떤]**[2] **[사용했던]**[3]? (That used as what?) |
| SKBPG | **[그것은]**[1] **[어떤 것으로]**[2] **[사용되었는가]**[3]? (What was that used as?) |
| TS | 우리 **[나라에서]**[1] **[최고로]**[2] 긴 다리는 **[길이가 얼마인가]**[3]? (What is the length of the longest bridge in our country) |
| Baseline | 우리 **[국가에서]**[1] **[많이]**[2] 긴 다리는 **[얼마인가]**[3]? (How much is the very long bridge in our country?) |
| SKBPG | 우리 **[국가의]**[1] **[가장]**[2] 긴 다리는 **[얼마나 긴가]**[3]? (How long is the longest bridge of our country?) |
| TS | 루이 암스트롱은 **[몇 년도에]**[1] **[출생하였는가]**[2]? (What year was Louis Armstrong born?) |
| Baseline | 루이 암스트롱은 **[시기는]**[1] **[태어났는가]**[2]? (Timeline was Louis Armstrong was born?) |
| SKBPG | 루이 암스트롱은 **[어느 년도에]**[1] **[태어났는가]**[2]? (In what year was Louis Armstrong born?) |

Table 1: Examples of test sentences (TS) and paraphrased sentences obtained using each method (Baseline and SKBPG). In the examples of sentences, the same superscript numbers indicate the source in a TS and the paraphrased phrase selected from each method.

trained together using the SVM to generate a paraphrase model. This model is used in the paraphrase generation phase, as described in section 3.2. Figure 3 illustrates the three types of features used in our model for one sentence from the bilingual parallel corpora example shown in Figure 1.

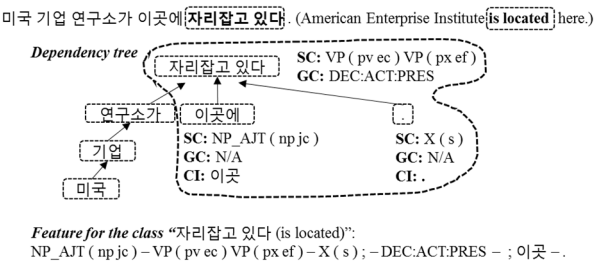


Figure 3: Sample sentence with the three types of features extracted.

### 3.2 Paraphrase Generation

In the paraphrase generation step, a source phrase in the input sentence is replaced by a candidate phrase in its corresponding paraphrase set, and as a result, a paraphrased sentence is produced. This step starts by locating a source phrase in an input sentence. For the input sentence "삼성 본사는 서울에 위치하고 있다. (The headquarters of Samsung is located in Seoul.)," as shown in Figure 1, our method first selects a source phrase. If multiple candidates appear, one with the maximum length of words is selected. If both phrases, "위치하고 있다 (is located)" and "위치하고 (located)" are possible candidates, for instance, the longer phrase "위치하고 있다 (is located)" will be selected. Next, dependent words preceding and following the source phrase are used together with the source phrase to obtain the three types of features described in section 3.1. Next, the SVM classifier is used to identify the best phrase in the paraphrase set for the source phrase that was built during the training phase.

Finally, the source phrase is substituted with the selected best paraphrase. Although in this example, we assumed the input sentence to have only one source phrase for simplicity, in our actual implementation the paraphrase generation process was repeated for multiple source phrases in the input sentence as shown in Table 1.

## 4 Evaluation

We evaluated our proposed method by comparing it with the popular method proposed by Bannard

and Callison-Burch (2005). This baseline[1] method was implemented by using probabilistic models and is described in section 2.1. Our proposed method, string kernel-based paraphrase generation (SKBPG), was implemented by using the edit distance string kernel, which is described in section 2.2.

### 4.1 Experimental Resources

In order to generate paraphrase sets, we used 998 randomly selected English sentences from the Text REtrievial Conference (TREC) question answering track (2003-2007)[2] and their translations (Korean words: 5,286, English words: 7,474). The question answering track was selected so that we could apply our method to a question answering system.

For the test sentences, 100 quiz sentences from Korean TV quiz shows (e.g., Golden Bell Challenge!) were selected. The sentences had to contain at least one possible source phrase with multiple candidates in its corresponding paraphrase set. Table 1 shows examples of the test sentences and the paraphrased sentences obtained using each method.

For the baseline method, we used 52,732 Korean sentences (Korean words: 322,306) in KAIST language resources (Choi, 2001) for training trigram LMs, in addition to the questions from TREC. This additional resource was included to make probability distribution in LM stable by expanding size of corpus. The LM probability was acquired using the IRSTLM toolkit (Federico and Cettolo, 2007), and conditional probability in LM was calculated by applying modified Kneser-Ney smoothing.

For the SKBPG method, we used ETRI linguistics analyzer (Lee and Jang, 2011) for dependency parsing and morphological analysis. For the SVM, we used LIBSVM-string (Guo-Xun Yuan, 2010; Chang and Lin, 2011), which supports the edit distance kernel option and multiclass classification based on the one-against-one approach, as described in section 2.2. The parameter of edit distance kernel ($\gamma$) was 0.1.

[1] We were not able to obtain Bannard and Callison-Burch's implementation, so we implemented it ourselves.

[2] These resources are available at http://trec.nist.gov/data/qa.html.

### 4.2 Evaluation Metrics

The Korean paraphrase pairs that we generated were evaluated by two native Korean speakers according to the following three criteria:

- **Meaning Preservation (MP):** Does a generated paraphrase preserve the meaning of the source phrase?
- **Grammaticality (G):** Is the generated paraphrase grammatical?
- **Equivalence (E):** Are the paraphrased pairs equivalent?

We used two types of scales as shown in Table 2. These criteria were adopted from previous research (Callison-Burch, 2008; Fujita et al., 2012).

| Criterion | 5-point | Binary scale |
|---|---|---|
| MP | (1: worst 5: best) | (true: MP > 3, false: otherwise) |
| G | (1: worst 5: best) | (true: G > 4, false: otherwise) |
| E | N/A | (true: MP > 3 & G > 4, false: otherwise) |

Table 2: Two types of scales used by the three evaluation criteria.

In terms of the inter-annotator agreement using Kappa, K = .412 for the 5-point scale, which is considered as "Moderate." For the binary scale, K = .612, which is regarded as "Substantial" (Landis and Koch, 1977; Carletta, 1996).

### 4.3 Results and Discussion

In the section, we summarize the results of our manual evaluation, which show that our method outperformed the baseline method, as shown in Table 3 and Table 4.

| | MP | G |
|---|---|---|
| Baseline | M = 3.28<br>SD = 1.29 | M = 3.54<br>SD = 1.23 |
| SKBPG | **M = 3.62**<br>**SD = 1.32** | **M = 3.97**<br>**SD = 1.20** |

Table 3: Results of the manual evaluation using the 5-point scale (M: mean, SD: standard deviation).

| | MP | G | E |
|---|---|---|---|
| Baseline | .57 | .42 | .36 |
| SKBPG | **.69** | **.58** | **.53** |

Table 4: Results of the manual evaluation using the binary scale.

For the manual evaluation using the 5-point scale, an independent-samples t-test showed that SKBPG significantly outperformed the baseline for both meaning preservation ($t(398) = 2.564$, $p = .011$) and grammaticality ($t(398) = 3.501$, $p = .001$). The evaluation using the binary scale also showed that SKBPG outperformed the baseline by 12%, 16%, and 17% for the three criteria of meaning preservation, grammaticality, and equivalence, respectively.

Interestingly, even though more resources were used in the baseline method for training the LM (52,732 Korean sentences), it did not outperform SKBPG. This suggests that our method is more efficient in terms of using fewer resources with less amount of data storage space. One plausible reason for such efficiency is that given that agglutinative languages have a large number of variants of lexicons for a root, it is difficult to account for most of the variations. Since the baseline method uses probabilistic models that utilize the frequency of each variation, much more data is needed. Another potential reason for the efficiency is that the word order for agglutinative languages is not critical for maintaining grammaticality. As opposed to isolating languages in which the word order determines grammatical functions, agglutinative languages use the postpositional particles or affixes of a root in a word to determine grammatical functions. Therefore, rather than using a LM that calculates the probability of contiguous words sequences, utilizing dependency grammar between words, as in SKBPG, can be more efficient.

## 5 Conclusion

In this study, we proposed a novel paraphrasing method, which considers the inherent characteristics of agglutinative languages by using an SVM with a string kernel.

Our evaluation of the generated paraphrases showed that the proposed method outperformed the probabilistic model-based method by 12%, 16%, and 17%, with respect to meaning preservation, grammaticality, and equivalence even with fewer resources than in the baseline method.

A limitation of this study is that the data set was aligned manually for paraphrase extraction between the two languages and due to this reason our data set size was relatively small with 1515 paraphrase sets. This limitation led to several problems in our evaluation. Sometimes, there were no appropriate grammatically correct candidates in the paraphrase sets for a certain input sentence. This also led to reduced coverage of paraphrases.

In addition, our method does not consider many semantic features such as semantic roles and named entities. This point suggests that our method is fragile in meaning preservation of the source sentence as the data size increases.

Therefore, we plan to work on automatic paraphrase extraction method tailored to agglutinative languages in order to increase the size of our data set. We also expect to expand the feature set by considering additional semantic features for our future work.

## Acknowledgments

This work was supported by the IT R&D program of MSIP/IITP [10044577, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services].

## References

Atsushi Fujita, Pierre Isabelle, and Roland Kuhn. 2012. Enlarging Paraphrase Collections through Generalization and Instantiation. In *Proceedings of EMNLP*, pages 631-642.

Changki Lee and Myung-Gil Jang. 2011. Large-Margin Training of Dependency Parsers Using Pegasos Algorithm. *ETRI Journal*, 32(3):486-489.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3): 1-27, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*, pages 196-205.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved Statistical Machine

Translation Using Paraphrases. In *Proceedings of HLT-NAACL*, pages 17-24.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of ACL*, pages 597-604.

Dekang Lin and Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343-360.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19-51.

Guo-Xun Yan. 2010. LIBSVM-String: An Extension to LIBSVM for Classifying String Data. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#libsvm_for_string_data.

Günes Erkan, Arzucan Özgür, and Dragomir R. Radev. 2007. Semi-Supervised Classification for Extracting Protein Interaction Sentences Using Dependency Parsing. In *Proceedings of EMNLP-CoNLL,* pages 228-237.

J. Richard Landis, Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics,* 33(1):159-174.

Jean Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249-254.

Kathlenn R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *Proceedings of HLT*. pages 280-285.

Key-Sun Choi. 2001. KAIST language resources v.2001. Result of Core Software Project from Ministry of Science and Technology, Korea (http://kibs.kaist.ac.kr).

Marcello Federico, and Mauro Cettolo. 2007. Efficient Handling of N-gram Language Models for Statistical Machine Translation. In *Proceedings of Second Workshop on StatMT*, pages 88-95.

Philipp Koehn, Franz josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL*, pages 48-54.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrase Generation as Monolingual Translation: Data and Evaluation. In *Proceedings of INLG*, pages 203-207.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-Driven Statistical Paraphrase Generation. In *Proceedings of ACL-AFNLP*, pages 834-842.

Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. 2007. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of ACL*, pages 464-471.

Zhiyang Wang, Yajuan Lü, Meng Sun, and Qun Liu. 2013. Stem Translation with Affix-Based Rule Selection for Agglutinative Languages. In *Proceedings of ACL*, pages 364-369.

Zornitsa Kozareva and Andrés Montoyo. 2006. Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. In *Proceedings of International Conference on NLP*, pages 524-533.