

Machine Translation Model using Inductive Logic Programming

Ahmad Hossny

Faculty of Computers and
Information, Cairo University
Cairo, Egypt

Ahmad.Hossny@ITDuty.com

Khaled Shaalan

Faculty of Informatics
British University in Dubai
Dubai, UAE

Khaled.Shaalan@buid.ac.ae

Aly Fahmy

Faculty of Computers and
Information, Cairo University
Cairo, Egypt

A.Fahmy@fci-cu.edu.eg

Abstract:

Rule based machine translation systems face different challenges in building the translation model in a form of transfer rules. Some of these problems require enormous human effort to state rules and their consistency. This is where different human linguists make different rules for the same sentence. A human linguist states rules to be understood by human rather than machines.

The proposed translation model (from Arabic to English) tackles the mentioned problem of building translation model. This model employs Inductive Logic Programming (ILP) to learn the language model from a set of example pairs acquired from parallel corpora and represent the language model in a rule-based format that maps Arabic sentence pattern to English sentence pattern.

By testing the model on a small set of data, it generated translation rules with logarithmic growing rate and with word error rate 11%

Keywords:

Machine translation; Inductive logic programming; rule based machine translation; example based machine translation; rule Induction; Arabic to English

1. Introduction

Machine translation systems usually depends on translation model that maps from source to target language, such mapping is implemented logically, statistically, example based or hybrids of them. But whatever the technique is it needs a set of resources to build the model that might be parallel corpora or parallel treebanks.

The motive to build such model is to tackle the lack of Arabic to English translation resources and the high cost of using linguists to build the needed resources as parallel corpora and syntactic structure and rule induction, plus the inconsistency of the human being output generally due to his natural probability for error [1].

Here we propose hybrid model that uses example based technique (Inductive Logic Programming) to build the translation model as transfer rules from Arabic phrase template to English phrase template.

The model consists of four steps (1) prepare training input parallel corpora a set of sentence pairs, (2) sentence partitioning (phrases) (3) morphological analysis for words to detect the semantic features of each word, (4) induce translation model rules by identifying LHS pattern, RHS pattern and transfer rule between them.

The translation process is done in five steps (1) analyzing input sentence's words, (2) find the most matching LHS pattern with the input, (3) produces the RHS pattern, (4) generate the target sentence words (5) concatenate generated words.

The rest of the paper is organized as follows. Section 2 explains briefly inductive logic programming concepts. Section 3 mentions related work for hybrid machine translation models. Section 4 explains the process of building translation model. Section 5 explains translation process using the generated model. Section 6, discusses the experimental results. Finally, concluding remarks and directions for future work are derived in Section 6.

2. Inductive Logic Programming

Inductive Logic Programming (ILP) [2] is a supervised machine learning technique that uses logic programming as a representation for examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system derives a hypothesized logic program which entails all the positive and none of the negative examples.

ILP work on two major steps (1) define the basic operations applied to the source example (machine learning input) to be converted to output example (machine learning output) where any further operation is a composition of such basic operations (2) train the system positively to identify which input should be mapped to which output as logical production rules (3)

train the system negatively to identify which inputs should never map to which output.

The main advantage of ILP generally is that it provides logarithmic growth (for first order learning [3] which leads to system consistency after limited set of trainings (degradation of newly generated rules), ILP is implemented in NLP domain like morphological analysis [4], Part-Of-Speech tagging [5] and POS disambiguation [6], but this is the first proposal of implementation in machine translation domain

3. Related Work

In this section we describe briefly the most famous machine translation hybrid models to describe other models integrated the example based techniques with rule based techniques or statistical techniques.

3.1. RBMT/EBMT Hybrids

Satoshi Shirai proposed by year 1997 a method that gathers the strength points in both of RBMT and EBMT for English-Japanese Translation. The algorithm was briefly described in three steps 1) Selecting a set of candidate sentences which are similar to the input sentence, 2) Selecting the most typical translation out of those corresponding to the candidates, and 3) Using this translation and its source as templates to translate the input sentence. By discarding candidates with atypical Translations, the algorithm filters out free, incorrect or context dependent translations [7].

In 1998 Michael Carl and others proposed an experimental and methodological objective to investigate the consequences of introducing an application-oriented Module (Case-Based Analysis and Generation, CBAG) component into a theoretically-based MT paradigm and to determine exactly what kind of linguistic entities (syntactic constructions, lexicographic types, collocations etc.) to be translated are suitable for simple uniform processing without entailing additional translation errors [8].

V.Ambati and Rohini proposed a system by 2007 for English-Indian machine translation system with high quality, they went to hybrid as the EBMT cannot handle the Indian language problems alone due to the small size of the parallel texts available and sparse linguistic resources and the word alignment problems in there, actually the main focus of such system was the Indian language challenges, and how it could be tackled using the stated techniques [9].

They have used the architecture proposed by P.Brown and others by 1993 [10] because it makes use of source and target parallel sentences. The advantage of this approach is that it can be language independent and makes minimal use of linguistic resources as they are

sparse in such case.

3.2. RBMT/SMT Hybrids

Y.Chen and others proposed an architecture that allows combining statistical machine translation (SMT) with rule-based machine translation (RBMT) in a multi-engine setup. It uses a variant of standard SMT technology to align translations from one or more RBMT systems with the source text. They incorporate phrases extracted from these alignments into the phrase table of the SMT system and use the open-source decoder MOSES to find good combinations of phrases from SMT training data with the phrases derived from RBMT[11].

Wu and Wang proposed an approach that first uses IBM statistical translation model to perform alignment in both directions (source to target and target to source), then uses the translation information in the rule-based machine translation system to improve the statistical word alignment [12]. The improved alignments allow the word(s) in the source language to be aligned to one or more words in the target language.

In 2008, Apptek built a hybrid MT system to integrate with Automatic Speech Recognition for media purposes by H. Sawaf in year 2008 [13]. Such system used the Rule Based Machine Translation as the main feed for the MT system, using Left Functional Grammar by Shihadah and Roochnik [14], as it contains a richly-annotated lexicon containing functional and semantic information. The system used the SMT to handle dialects and variations (Modern Standard Arabic variation) because SMT better handles phrases and maximum entropy (Combine different knowledge sources while analyzing sentence for translation).

3.3. EBMT/SMT Hybrids

K.Imamura proposed an example based MT method based on syntactic transfer, which selects the best translation by using models of statistical MT. This method is roughly structured using two modules. First module is an example-based syntactic transfer module which constructs tree structures of the target language by parsing and mapping the input sentence while referring to transfer rules. The other module is a statistical generator, which selects the best word sequence of the target language in the same manner as statistical MT. Therefore, this method sequentially combines example-based and statistical MT. The proposed method has the following advantages[15].

By 2004, E.Sumita proposed a project called Corpus-Centered Computation (C3). C3 places corpora at the center of its technology, translation knowledge is extracted from corpora, translation quality is gauged by referring to corpora, the best translation among multiple-engine outputs is selected based on corpora, and

the corpora themselves are paraphrased or filtered by automated processes to improve the data quality on which translation engines are based. This proposes two endeavors that are independent: a) a hybridization of EBMT and statistical models, and b) a new approach for SMT, phrase-based HMM. The hybridization was used in the “unrestricted” Japanese-to-English track, while the phrase-based HMM was used in “supplied” Japanese-to-English and Chinese-to-English tracks [16].

Eiji Aramakil and others proposes in 2005 a probabilistic translation model, which deals not only with the example size but with the context similarity. In the experiments, the proposed model has a slightly better translation quality than the state of the art EBMT systems. The results demonstrated the validity of the proposed model. The Algorithm of proposed method consists of the following two modules: (1) an alignment module, which builds translation example from corpus through a sequence of three steps (a) conversion into phrasal dependency structures, (b) alignment based on translation dictionary (c) building translation example database, and (2) a translation module, which generates a translation through three steps a) input sentence analysis (b) select translation examples (c) Output sentence generation [17].

4. Building Translation Model

Translation model is the knowledge base that is used to transform source sentence to target sentence in the runtime translation process. E.g. BNF rules in RBMT and statistical model in SMT. The proposed model depends mainly on rules, it is built in four major steps, preparing data, sentence partitioning (chunking) and morphological analysis which are kind of preprocessing, then induce translation rules and set unification variables to build the language model depending on the prepared data.

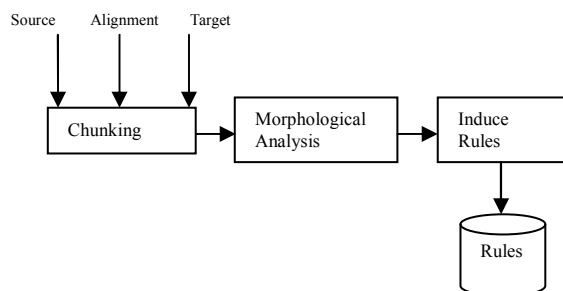


Figure 1: Building translation model.

4.1. Prepare training inputs

It is set of example pairs (sentence and its translation) with word alignment between them (the alignment should be defined before training).

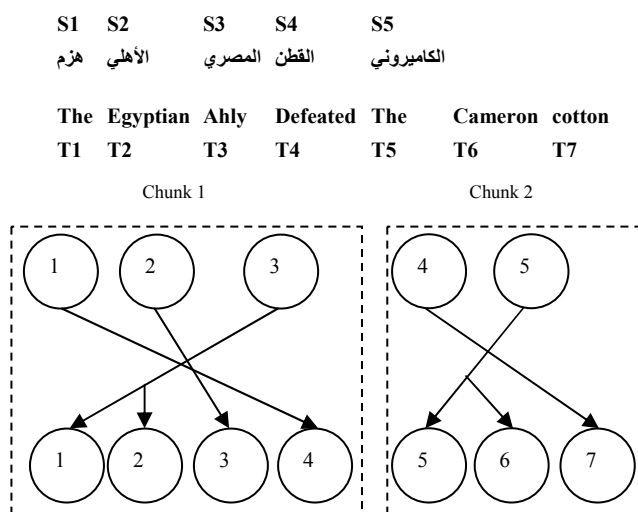


Figure 2: Chunking based on word alignment.

4.2. Partition the Sentence (Chunking)

Since we can't deal with the whole sentence as once due to complexity factor we have to Partition both source and target sentence sequence of chunks. This is done according to word alignment dependency, such that there is no overlapping of word ordering between chunks. This is declared in figure 2.

To detect the chunk boundaries we made simple algorithm (the point of splitting the sentence into chunks) which is an implementation of phrase extraction heuristics mentioned by P.Koehn [18].

Algorithm: *FindChunkBoundary*
 Inputs: Source Sentence S_s , Target Sentence T_s ,
 Source Chunk End Index $source_end$,
 Target Chunk End Index $target_end$

Parameters default values:
 $source_end = source\ chunk\ start\ location$
 $target_end = target\ chunk\ start\ location$

Output: Index of the end of the chunk

Algorithm:

1. $src_max_idx \leftarrow \max(\text{index of } source_end)$
2. For each word $target_word$ in target sentence T_s let $target_idx$ be the index of $target_word$ (where the $target_idx$ is less than $target_end$)
 - a. $src_idx \leftarrow \text{index (word associated with } target_word \text{ in source sentence)}$
 - b. If $Src_max_idx < src_idx$ let $Src_max_idx \leftarrow src_idx$
3. $target_end\ word \leftarrow \text{word at the } target_end$

```

position in the target sentence  $T_s$ 
4. If ( $src\_max\_idx > source\_end$ )
    a.  $new\_target\_end \leftarrow FindChunkBoundary(T_s, S_s, target\_end, src\_max\_idx)$ 

    b. If  $new\_target\_end == target\_end$ , return  $src\_max\_idx$  as the source chunk boundary.

    c. Else return  $new\_src\_idx$  as the index of the word associated with  $new\_target\_word$  in source sentence

5. Else return  $source\_end$ 

```

Algorithm 1; partitioning sentence based on word alignment (sentence chunking)

4.3. Morphological Analysis:

Analyze each word morphologically in the source sentence and target sentence using morphological analyzer AMRIS to get the feature vector of each word the represents its semantic properties like gender, count, tense, etc. [5].

This leads to having a sentence that consists of sequence of chunks; each chunk consists of sequence of morphologically analyzed words with its semantic prosperities.

4.4. Building Translation Rules

To induce the translation rules using ILP we need to analyze the source chunk and the target chunk and the transformation done to convert source to target.

This analysis is done in two steps (1) detect word level operations, (2) build the rule using these operations to identify LHS mapped to RHS, then we unify the semantic features detected by morphological analysis between each of LHs and RHS .

So, we will first state the possible operations to be applied to any word in the source chunk to be transformed to its target chunk. Then for each chunk analyze the applied operations, and then build the transfer rules describing the pattern and features of source and target chunks.

a. Translation basic operations that applies to most sentences :

- i. **Direct Transfer:** were no changes is done to the word structure:
 - English: *Mohammed went to School (LTR)*
 - Arabic: *محمد ذهب إلى مدرسة (RTL)*
- ii. **Merge:** were two words (source language) are mapped to one word (target language) because either

morphological change or the richness of target word:

- English: *The police arrested the criminal (LTR)*
- Arabic: *الشرطة أمسكت المجرم (RTL)*

The Police → الشرطة
Blood Feud → الثأر

iii. **Split:** were one word (source language) is mapped to two words (target language) either for morphological effect in source language or for the meaning itself:

- English: *Mohammed came home at midnight (LTR)*
- Arabic: *محمد عاد للمنزل في منتصف الليل (RTL)*

Midnight → منتصف الليل
Fired → أطلق النار

iv. **Phrasal Words:** when a word changes its meaning according to its following preposition (so word + prep. Are treated as a whole) , and could be mapped to one word or more in target language:

- English: *the mother look after her children (LTR)*
- Arabic: *الأم تعتني بأطفالها (RTL)*

Look After → تعتني

v. **Idioms:** were an expression in source language mapped to an explaining sentence in target language or mapped to other expression in target language:

- English: *It rains cats and dogs (LTR)*
- Arabic: *إنها تمطر بغزارة (RTL)*

Rains cats and Dogs → تمطر بغزارة

These word operations are used to identify mapping rules from source chunk pattern to target chunk pattern, where the mapping rule actually defines the relocation of words inside such chunk with words that apply previously mentioned operations

As we need to detect the mentioned operations applied to each word in the input chunk and output chunk. We defined the following algorithm that utilizes the alignment between both chunks from the training data:

```

Algorithm: DetectWordsOperations
Input: source chunk, target chunk, alignment
Output : set every word's operation in the source chunk that maps to target sentence

If source chunk is idiom
    For each word  $W_s \in$  source chunk  $S$ 
         $W_s.operation = 'idiom'$ 
         $W_s.target =$  index of all  $W_t$  such that  $W_t \in$  target chunk
    Else

```

For each Word W_s such that $W_s \in$ source chunk S , and m, n are different indexes in the source chunk and p is index of word in target chunk

If $W_s[m].targetwords.count > 1$
 $W_s.operation = 'split'$

If $W_s[m].targetword == W_t[p]$ and
 $W_s[n].targetword == W_t[p]$

$W_s[m].operation = 'merge'$
 $W_s[n].operation = 'merge'$

If $W_s[m]$ and $W_s[n]$ forms Phrasal Word
 $W_s[m].operation = 'phrasal'$
 $W_s[n].operation = 'phrasal'$

Algorithm 2; detecting word operations per each chunk

b. Construct mapping rules using defined translation operations:

Rule construction process is done by stating the RHS as sequence words and with their operations and feature values and LHS as sequence of words and their feature values then apply unification between the two sides.

The unification between is done using Unification Based Grammar representation as setting specific values (represented in small letters) and unified variables (capital starting letters) and don't care values ($_$).

So we identify LHS and RHS using (1) word operations (2) semantic features detected in the morphological analysis phase, and then unify both sides together.

The following example represents the process of rule construction:

Example ;

Chunk1= المصري , الأهلي , هزم = هزم الأهلي المصري
 Chunk2= The Egyptian al-Ahly defeated

Words mapping and morphological analysis will be:

هزم	→	defeated
Wa[1] (direct transfer)	→	We[1]
{count:single, sex:male, type:verb, tense:past}	→	{count:single, sex:male, type:verb, tense:past}
{count:S1, sex:male, type:verb, tense:PI}	→	{count:S1, sex:_ , type:verb, tense:PI}

الأهلي	→	al-Ahly
Wa[2] (direct transfer)	→	We[2]
{count:single, sex:_ , type:propernoun}	→	{count:single, sex:_ , type:propernoun}
{count:S2, sex:_ , type:propernoun}	→	{count:S2, sex:_ , type:propernoun}

المصري	→	Egyptian
Wa[3] (split)	→	The + We[3] + We[4]
{count:single, sex:male, type:adjective, defined:al}	→	{type:preposition} + {count:single, sex:male ,
{count:C3, sex:S3, type:T3, defined:al}	→	{type:preposition} + {count:C3, sex:_ , type:T3}

The resulting rule structure is:

Wa[1] + Wa[2] + Wa[3] → The We[4] + We[2] + We[1] this defines the target order of words

By including the morphological rules and applying unification for LHS and RHS, it results the rules:

```

Wa[1] {count:S1, sex:male, type:verb, tense:PI} +
Wa[2] {count:S2, sex:_, type:propornoun} +
Wa[3] {count:C3, sex:S3, type:T3, defined:al}
→
The {type:preposition} We[4] {count:C3, sex:_, type:T3} +
We[2] {count:S2, sex:_, type:propornoun} +
We[1] {count:S1, sex:_, type:verb, tense:PI}

```

Where , capital letters means unified variables , and small letters means specific values, underscore means Don't Care value

Example 1: Rule generation process (step to build translation model)

5. Translation Process

Translation process consists of 5 main steps that utilize the previously built mode; these steps are displayed in the following figure 2.

The following steps describe briefly the runtime translation process.

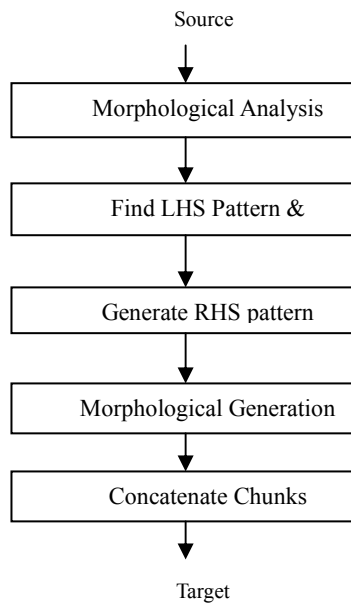


Figure 2: Translation process

- Analyze each word in the input sentence morphologically to get the feature vector that contains the semantic description
- Find the LHS pattern that matches the whole sentence structure from the rule base previously induced. If no LHS matches the whole sentence, partition the sentence into two or more chunks such that each chunk will match with some LHS of some rule. In case of multiple rules matching the

same sentence or chunk, we get the rule with highest probability.

- Apply the rule to produce the RHS for the target language sentence structure, taking into consideration unified variables between RHS and LHS.
- Generate target words morphologically for each chunk to apply the semantic features set by the production rule.
- Concatenate different chunks to compose the complete sentence.

6. Experimental results

The proposed model have been tested on two phases, first is to build the language model to generate transfer rules. This process uses dataset of size 300 sentences with average length of 10 words selected from (LDC2005E46). By applying chunking algorithm, it produced 2130 chunks with distribution in Table 1.

Chunk size	Chunk Occurrences
1	1498
2	304
3	151
4	81
5	32
6-7	15
8-9-10	6

Table 1: Chunks occurrences

These chunks are used to induce rules that will be used for translation process. We have tested the rule induction process with different number of samples, where the repeated rules are not considered, it results 1115 transfer rules, the important part is that growing

rate have taken logarithmic shape (Figure 3). This leads that for a specific number of examples we can induce the most frequently used unique rules for Arabic to English transfer.

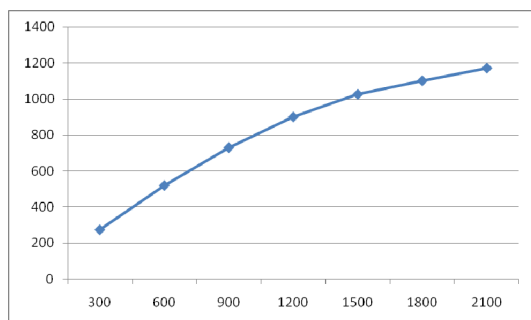


Figure 3: induced rules growing rate

The Translation process have been tested using the 300 sentence from the training set used for rule induction, and induced rules covered 97% of the sentences with ambiguity rate 8%, and by applying word error rate evaluation (WER) that implements Levenestien formula [19] on sentence level rather word level it gave .04 error rate.

By testing the translation process using 180 other sentences from the same dataset (LDC2005E46), induced rules covered 93% of the input sentences (source language), with ambiguity rate 10%, and by applying WER evaluation it gave 0.11 error rate.

7. Conclusion and future work

The proposed model is a proof of concept that ILP can be used to build rule based system and fix the problems of rules extraction effort and rules consistency that result from manual work.

The results of such integration grow in logarithmic way that leads to system saturation after specific number of training examples. It is kind of hybrid system where we utilize the example based technique to build rule based translation model (pattern transfer).

The results of the experiment is promising on the small scale, the future challenge is to test it on very large scale hundreds of thousands sentences to build real time system. This model has many future advances that enhance the performance and achieve the translation systems targets, the most important ones are:

- Build negative rule base through wrong examples to exclude totally denied rules especially in formal language representation.
- Handle special cases of language translation like idioms, collocation, phrasal verbs, etc.
- Integration with statistical translation techniques by applying n-gram technique or with different fertility levels.
- Apply ambiguity resolution techniques.

References

- [1] Eiichiro Sumita, Hitoshi Iida: Experiments and Prospects of Example-Based Machine Translation. ACL 1991: 185-192
- [2] S.Muggleton: Inductive Logic Programming: Issues, Results and the Challenge of Learning Language in Logic, Artificial Intelligence (1999), 114(1-2), pp. 283--296.
- [3] F.Koriche, (2005). Online closure based learning of relational theories, ILP 2005, pp. 172--189.
- [4] A.Hossny, K.Shaalan, A.Fahmy. automatic morphological rule induction (LREC 2008), Arabic Workshop.
- [5] M.Einborg, N.Lindberg,(1999): ILP in Part-Of-Speech Tagging – An Overview. Learning Language in Logic: 57 – 169.
- [6] N.Lindberg , M.Einborg,(1999): Improving Part Of Speech .disambiguation rules by adding linguistic knowledge. ILP999: 186-197.
- [7] Satoshi Shirai, Francis Bond, Yamato Takahashi : A Hybrid Rule and Example-based Method for Machine Translation , (Recent Advances in Example-Based Machine Translation, 1997)
- [8] Michael Carl, Catherine Pease, Leonid L. Iomdin, Oliver Streiter: Towards dynamic linkage of ExampleBased and Rule-Based Machine Translation (ESSLLI '98 Machine Translation Workshop)
- [9] Vamshi Ambati, Rohini U : A Hybrid Approach to Example based Machine Translation for Indian Languages (ICON 2007)
- [10] Peter F. Brown , Vincent J. Della Pietra , Stephen A. Della Pietra , Robert L. Mercer, The mathematics of statistical machine translation: parameter estimation, Computational Linguistics, v.19 n.2, June 1993
- [11] Yu Chen1, Andreas Eisele1,2, Christian Federmann2, Eva Hasler3, Michael Jellinghaus1, Silke Theison1,: Multi-Engine Machine Translation with an Open-Source Decoder for Statistical Machine Translation (ACL 2007)

- [12] WU Hua, WANG Haifeng: Improving Statistical Word Alignment with a Rule-Based Machine Translation System (International Conference On Computational Linguistic , 2004)
- [13] H.Sawaf, B.Gaskill, M.Veronis : Hybrid Machine Translation Applied to Media Monitoring , (AMTA-2008)
- [14] Shihadah, M., P. Roochnik.: Lexical-Functional Grammar as a Computational-Linguistic Underpinning to Arabic Machine Translation. In Proceedings of the 6th International Conference and Exhibition on Multi-lingual Computing, Cambridge, UK, 1998.
- [15] Kenji Imamura, Hideo Okuma, TaroWatanabe, and Eiichiro Sumita : Example-based Machine Translation Based on Syntactic Transfer with Statistical Models (International Conference On Computational Linguistics , 2004)
- [16] Eiichiro SUMITA, Yasuhiro AKIBA, Takao DOI, Andrew FINCH, Kenji IMAMURA, Hideo OKUMA, Michael PAUL, Mitsuo SHIMOHATA, Taro WATANABE :EBMT, SMT, Hybrid and More: ATR Spoken Language Translation System (Proceeding of International Workshop on Spoken Language Translation, Kyoto, Japan. 204)
- [17] Eiji Aramaki, Sadao Kurohashi², Hideki Kashioka³ and Naoto Kato⁴: Probabilistic Model for Example-based Machine Translation (Proceedings of MT Summit X, 2005)
- [18] Philipp Koehn, Franz Josef Och, and Daniel Marcu, Statistical Phrase Based Translation, HLT / NAACL 2003.
- [19] Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady, 10(8), pp. 707--710.