Experiments with String Similarity Measures in the EBMT Framework

Natalia Elita, Monica Gavrila, Cristina Vertan
Faculty of Mathematics, Informatics and Natural Sciences,
Department of Informatics, University of Hamburg
{*elita, gavrila, vertan*}@*informatik.uni-hamburg.de*

Abstract

Measuring string similarity is a frequently used technique in various Language Technology (LT) applications, such as: Spell checkers, Translation Memories, Example-Based Machine Translation (EBMT) etc.

In this paper experimental results on string similarity measures are presented. The main goal of the experiments is to detect the most appropriate similarity measure which can be applied for retrieving candidate sentences for translation templates to be used in an EBMT system. The advantage of this approach is that it is based entirely on surface forms, therefore being independent from any linguistic resources. The results show that token-based measures are the most suitable for translation template extraction.

Keywords

String Similarity Measures, EBMT, Overlap Coefficient

1 Motivation

Measuring string similarity is a frequently used technique in various Language Technology (LT) applications, such as: Spell checkers, Translation Memories, cognates extraction from bilingual texts, sentence and word alignment, Example-Based Machine Translation (EBMT) etc. In this section the motivation to use string similarity measures in the EBMT framework is addressed.

Machine Translation (MT) - translation from one natural language into another by means of a computerized system, (see [1, 6, 5] for more details) - is a task of Natural Language Processing (NLP) that is being continuously studied and many attempts have been made to improve the quality of its output.

There are several approaches to the MT (e.g. rule-based MT, statistical MT etc.). The current paper focuses on the EBMT approach, that was first inspired by Makoto Nagao ([8]). EBMT is an implementation of the translation by analogy principle, which states that humans translate by first decomposing a sentence into sub-phrases, translating these sub-phrases, which are then combined into a translation of a given sentence. A part of any EBMT system is a database of examples, that can be stored as: strings, annotated tree structures, generalized examples (templates), etc. In this paper the template-based EBMT is chosen as a framework of the present research. In order to get a translation for a given string, three stages have to be performed. First, matching the input on the database of templates, then retrieving the corresponding target language (TL) parts and finally recombining the TL parts into a coherent translation (for details about EBMT in general, and template-based EBMT in particular refer to [4, 7, 10, 11]).

In EBMT similarity measures are used in the matching phase: given an input string in the source language (SL), similar sentences from the database of examples are retrieved, by means of a given similarity measure. In this paper similarity measures are used in the process of building the translation templates. This is realized by means of a Similarity Matrix (defined below), that uses the similarity measures in order to find good candidate sentences (see Example 1), which would later be generalized into templates.

The motivation for such a research comes from the problems encoutered while generalizing pairs of sentences into templates, as outlined in [7]. The algorithm used, namely the principle of string co-occurrence and frequency thresholds, states: SL and TL strings that co-occur in two (or more) sentence pairs of a bilingual corpus are likely to be translations of each other.

Example 1: Given two entries of an English-German corpus
1: <en>Construction of research reactor at Garching underway</en> –> <de>Startschuss fuer Bau des Forschungsreaktors in Garching</de>
2: <en>Accompanied by protests , the first sod was turned today for the construction of the new nuclear research reactor .</en> –> <de>Begleitet von Protesten ist heute der Startschuss fuer den Bau des neuen Forschungsreaktors bei Muenchen gefallen .</de>

In the SL part the strings that co-occur are: *construction*, *of*, *research*, *reactor*; in the TL part: *Startschuss*, *fuer*, *Bau*, *des*, *Forschungsreaktors*.

Thus, the two sentences can be generalized into a template of the form:
[construction of research reactor **V1**] – [Startschuss fuer Bau des Forschungsreaktors **V11**], or
[**V1** construction of **V2** research reactor **V3**] – [**V11** Startschuss fuer **V21** Bau des **V31** Forschungsreaktors **V41**], where **Vi** corresponds to a variable in the template.

Hence, the two sentences are good candidates for templates: they are **similar enough** (see section **3**). Similarity is calculated on surface-forms only, there-

fore the use of any linguistic resources is unnecessary.

Definition: For a monolingual corpus with N sentences, the **similarity matrix** S is defined as follows:

$$s(i,j) = -1,\ for\ j < i, 1 <= i, j <= N;$$
$$s(i,i) = 1,\ for\ 1 <= i = N$$
$$s(i,j) = BSM(sentence_i, sentence_j),\ \text{for}$$
$$j > i, 1 <= i, j <= N;$$
$$where\ BSM = Best\ Similarity\ Measure$$

As sentence similarity is a symmetric property, only values above the main diagonal are examined.

The advantage of using the similarity matrix is that only a sub-corpus, created from these sentences being above a certain threshold, is used as input for the template extraction engine, thus the search space for templates is considerably limited. The thresholds are experimentally determined, as shown in section **3**.

Twenty-one similarity measures were analyzed and compared. Those ones performing best were used to build the similarity matrix for a given SL (cf. Figure 1).

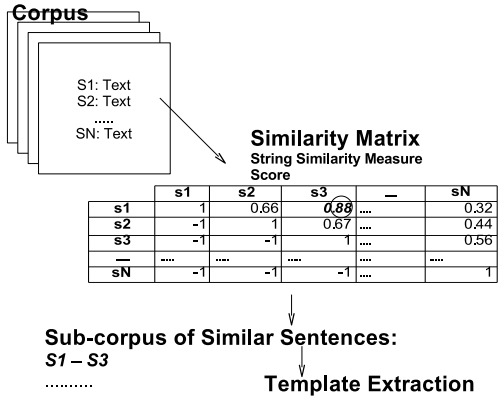

Fig. 1: *Extracting similar sentences*

The rest of the paper is organized as follows: in the next section, two modified similarity measures are described and the definitions of the measures used to create the similarity matrix are introduced. An account of the results obtained from a series of experiments made on string similarity measures is given in the third section. Finally the conclusion and further work are presented.

2 Similarity Measures

String similarity measures are divided in the literature into three categories: character-based, token-based, and hybrid. In the case of the first two, the similarity is calculated on character and token level respectively. In the case of the hybrid measures, the similarity is first calculated on the character level, then the obtained scores are used by a token-based metric. A good definition, purpose, and classification of similarity measures can be found in [3].

In the experiments, twenty-one similarity measures of all three types mentioned above were investigated. Eighteen of these measures are part of the SimMetrics package (SimMetrics is an open source Java library of similarity measures. For more details refer to [9]). Additionally we modified and extended two of them (**Normalized Token Levenshtein Distance** and **Longest Common Subsequence Similarity**) and implemented one (**Common Words**), for the purpose of finding similar sentences.

2.1 Modified Similarity Measures

Normalized Token Levenshtein Distance (**NTLD**) is a modified version of the traditional character-based Levenshtein Distance, and it has the following formula:

$$NTLD(s1, s2) = 1 - \frac{TLD}{2 * max(Length(s1), Length(s2))},$$

where TLD is the traditional Levenshtein Distance, but at token level, and $Length(s)$ means the number of tokens of s.

The **Longest Common Subsequence Similarity** (**LCSS**) is based on the Longest Common Subsequence (LCS) character-based algorithm. More details on this algorithm can be found in [2]. The initial algorithm is transformed into a token-based one. This way the token-level LCS between two sentences is computed. Given two sentences $s1$, and $s2$ the computation of the LCSS follows the steps below:

1. Calculation of the LCS at token level:

$$LCS_{TokenString}(s1, s2) = LCS\_String$$

2. Calculation of the LCSS at token level as:

$$LCSS_{Tokens}(s1, s2) = \frac{Length_{token}(LCS\_String)}{max(Length_{token}(s1), Length_{token}(s2))}$$

3. Subtraction of a penalty of 0.1 for each word-distance, in case the words found in the LCS_String are not one after another in the sentences $s1$, $s2$. In case of multiple results, the maximum value is considered. This score is $LCSS_{Penalties}$.

4. If the output of step **3.** contains multiple results, the longest one (as number of characters), is considered as best the match. The computation is done according to a formula similar to the one in step **2**:

$$LCSS_{Characters}(s1, s2) = \frac{Length_{characters}(LCS\_String)}{max(Length_{characters}(s1), Length_{characters}(s2))}$$

LCSS takes values within $[0, 1]$. 0 indicates that the sentences are completely different, and 1 that the sentences are identical.

2.2 Other Similarity Measures

In this subsection the definitions of the measures used to build the similarity matrix are presented.

Common Words (**CW**) is a trivial similarity measure that counts the number of identical tokens

(words) for the two given strings. It does not take into account the word order.

Overlap Coefficient (**OC**) is a metric that determines to what degree one string is a substring of another. Its formula is given below:

$$OC(s1, s2) = \frac{(|s1\&s2|)}{min(|s1|, |s2|)},$$

where $|s|$ is the number of tokens in s, and $|s1\&s2|$ the number of common tokens in $s1$ and $s2$.

3 Experimental Results

In this section the experiments we made in order to find similar sentences for template extraction are described and some of their results are presented. Two parallel, sentence aligned corpora were used for the experiments:

- a technical corpus in three languages: German (Ge), Romanian (Ro), and English (En), of approximately 2300 sentences (cca. 25000 tokens for each language);
- a small news corpus of 100 sentences, in German and English

First, the thresholds for each similarity measure were experimentally determined. Then a decision was made on which of the considered similarity measures is more effective for the goal that was set.

For each similarity measure, the initial threshold was established after testing the measure on a small set of artificial examples. Observations were made on how the value changed when the compared sentences were of different length, when the word order was different etc. This value was adjusted afterwards, as a result of testing each measure on the real data, namely, 100 sentences of a corpus, so that the precision increases.

Measure	**Threshold Value**
Common Words (CW)	initial 5, modified 3
NTLD	0.7
Matching Coefficient	0.55
Block Distance	0.6
Jaccard Similarity	0.45
Overlap Coefficient (OC)	initial 0.66, modified 0.5
Q-Grams Distance	0.65

Table 1: *Token-based similarity measures with the established thresholds*

In Table 3 **CONC** means the Chapman Ordered Name Compound Similarity. More details on the measures can be found in [9].

A threshold is a minimal value calculated for two similar sentences. A pair of sentences in a SL is considered to be **similar enough**, when the sentences under consideration fulfill the following constraints:

1. have at least three words in common (**CW Threshold**);
2. the sequence of common elements consists of at least 50% content words (lexical words);
3. one sentence is a sub-sentence of the other to the proportion of 50% (**OC Threshold**).

Measure	**Threshold Value**
TagLink Token	0.5
Euclidean Distance	0.5
Smith-Waterman (**SW**)	0.6
Smith-Waterman-Gatoh	0.6
Jaro	0.7
Jaro Winkler	0.7
Needlemann-Wunch	0.7
Levenshtein Distance	0.75,
Dice Similarity	0.75,
Cosine Similarity	0.75

Table 2: *Character-based similarity measures with the established thresholds*

Measure	**Threshold Value**
Monge-Elkan	0.9
CONC Similarity	0.75
TagLink	0.7

Table 3: *Hybrid similarity measures with the established thresholds*

The closer the value to 1, the more similar the sentences are. The value of 0 indicates that the sentences are completely different, and the value of 1 indicates that the sentences are identical. In the tables 1, 2, 3 an overview of the similarity measures with the established thresholds is given.

In the first experiment, similar sentences were extracted from 100 sentences taken from the technical corpus. This small number of sentences was chosen for an easier interpretation of the results, and in order to make observations and assumptions. The results are reflected in Table 4.

The experiments were run on each of the three categories of measures mentioned in section **2**. As a result, the same sentence pairs were extracted by several similarity measures of the same category. That is why the total number of sentences and the unique number are different.

The following observations and conclusions were drawn from the analysis of these data. From each group of similarity measures, the one that extracts the most similar sentence pairs that would be best candidates for the template extraction is chosen.

Hybrid methods seemed the most promising in theory, but proved to be not efficient in practice. From this group, TagLink measure, though it extracted a relatively small number of sentence pairs, was chosen as the best.

Example of sentence-pair extracted: - English. *TagLink: 0.76*
Writing and sending a multimedia message
Reading and replying to a multimedia message

Although the **character-based measures** extract the biggest number of sentence pairs, they depend a lot on the length of the sentences. They generally are not suitable for the EBMT. A good example is given in [11]. They proved to be quite slow and ineffective for the goal that was set. The sentence-pairs they

Token-based	**Ge**	**En**	**Ro**
CW	4	11	11
Matching coefficient	12	10	9
Block Distance	13	12	13
Jaccard Similarity	12	10	9
OC	24	19	25
Q-Grams Distance	9	9	6
Total	74	71	73
Unique pairs	26	30	31
Character-based	**Ge**	**En**	**Ro**
Levenshtein Distance	1	3	2
Dice Similarity	5	4	3
Cosine Similarity	5	4	3
Euclidean Distance	5	4	3
Jaro	35	32	56
Jaro-Winkler	86	72	109
Needleman-Wunch	24	40	22
SW	83	82	49
SW-Gotoh	107	103	73
Tag Link Token	70	67	62
Total	421	411	382
Hybrid	**Ge**	**En**	**Ro**
CONC	48	48	29
Tag Link	19	17	19
Total	67	65	48
Unique pairs	58	59	40

Table 4: *Number of sentence pairs extracted by each similarity measure*

extracted were not similar enough to be good candidates for translation templates. Some of the extracted sentence-pairs had in common only some characters. Smith-Waterman-Gotoh extracted the biggest number of sentence pairs in case of German and English, and Jaro-Winkler in case of Romanian.

Example of sentence-pair extracted: - German. *SmithWatermanGatoh: 0.6*
Kurzmitteilungen
Lesen und Beantworten einer Multimedia - Mitteilung

Token-based similarity measures proved to be the most effective for the goal.

Example of sentence-pair extracted: *CW+OC*
German
Einstellungen fuer Kurzmitteilungen und E-mail - Mitteilungen
Einstellungen fuer Multimedia - Mitteilungen
English
Settings for text and e-mail messages
Settings for the multimedia messages
Romanian
Setari pentru mesaje text si e-mail
Setari pentru mesaje multimedia

The **OC** measure performs the best (highest number of **similar enough** sentences) of all for all the three languages considered. However, considering the type of the corpus, a disadvantage was noticed: **OC** extracts many sentence pairs, where only one or two tokens overlap. This way the length of a possible template is too short. It happens especially in the case when one of the two sentences to be compared is very short, and is totally contained in the other sentence.

This disadvantage can be easily overcome, if **CW**, with an established threshold is used on the set of sentence pairs extracted by the **OC**. When combined, the thresholds were set to 3 for **CW**, and 0.5 for **OC**.

The results of **OC** combined with **CW** (**OC+CW**) were compared with the outcome of the **NTLD** and of **LCSS** combined with **CW** (**LCSS+CW**). The threshold for the **LCSS** was established at 0.33 and for **NTLD** at 0.7.

Experiments were run on the same set of 100 sentences. The results are included in Table 5.

	German	**English**	**Romanian**
OC + CW	18	31	27
NTLD	16	39	32
LCSS + CW	14	34	23

Table 5: *Sentence pairs above the established thresholds*

It can be noticed that quantitatively the results are comparable, but qualitatively they differ a lot. The **NTLD** extracts many sentences, where only short sequences overlap. The quality of the sentence pairs extracted by **OC+CW** is higher. Thus these sentence pairs become better candidates for templates. The number of extracted sentence pairs in German is smaller. This can be conditioned by the structural peculiarity of the language, and by the fact that the algorithms are case-sensitive for this language.

LCSS extracts valid pairs if combined with **CW**, having the same threshold as in the case of **OC**: 3. Unlike **OC+CW**, **LCSS+CW** considers also the word order of the two compared sentences.

Further, the precision and the recall of the best similarity measures, namely **OC+CW** were computed.

The results are included in table 6.

	German	**English**	**Romanian**
Precision	1	$0,7$	$0,96$
Recall	1	1	1

Table 6: *Precision and recall calculated on 100 sentences*

The value of recall is always one, as the first and third constraints from the **similar enough** sentences definition (Section **3**) were taken into account.

3.1 Other Experiments

In this subsection two new experiments are described: the first shows how the number of the extracted similar sentences is influenced by the language (language dependency), the second by the corpus type (corpus dependency).

1. The combination of **OC** with threshold set to 0.5 and **CW** set to 3 was used to build the similarity matrix as this combination proved to be the most effective for the goal. It was built for sets of different size in different languages for the technical corpus (cf. Figure 2). The chart shows us that a comparable number of similar sentence pairs is extracted for English and German, as for Romanian - a smaller number, compared to English and German. Two reasons can explain this outcome:

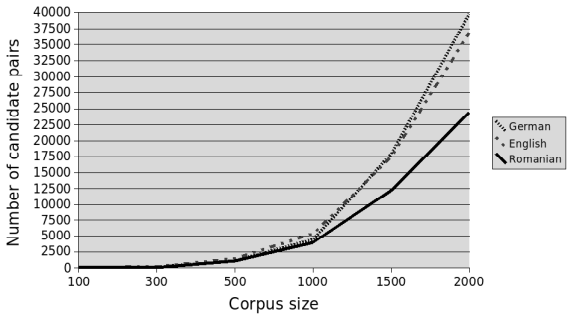

Fig. 2: *Sentence pairs extracted - Technical Corpus*

- German and English are both Germanic languages, while Romanian is a representative of the Romance languages;
- Compared to the other two languages, Romanian is a highly inflected language, especially in case of nouns and adjectives (e.g. the Romanian word '*lampa*' - English '*the lamp*' - has six (6) inflected forms).

2. An experiment with different type of corpora was made to check how corpus dependent the amount of extracted sentence pairs is. The results of the experiment run on 100 sentences corpora (technical and news) are shown in figures 3 and 4.

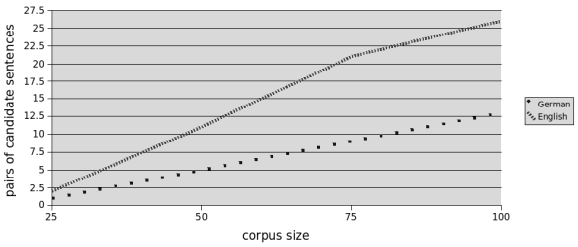

Fig. 3: *Sentence pairs extracted - News Corpus*

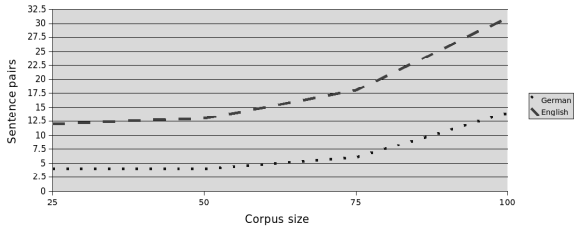

Fig. 4: *Sentence pairs extracted - Technical Corpus*

From these charts, one can see that the shape of the curves for the two languages is rather similar in the case of the technical corpus, and slightly different for the news corpus. A bigger number of sentence pairs is extracted for the technical corpus due to its restricted language.

A smaller number of sentence pairs is extracted for German in both cases. One of the reason is the value of the **CW** threshold, which is set to 3. A language specific characteristic for German is the composition of words, which correspond to several words in English: e.g in English: '*the tax reform*' reaches the threshold, but its correspondent in German: '*die Steuerreform*' is below the threshold. This proves that, in order not to lose data, the thresholds should be language-sensitive.

4 Conclusions

In this paper a comparison of string similarity measures in the framework of EBMT is presented. A similarity matrix is defined and used to find similar sentence pairs, that become candidates for translation templates. Twenty-one string similarity measures were analysed, including two modified similarity measures. Experiments were run on two sets of data in three languages. When building the similarity matrix a combination of **CW** and **OC**, or of **LCSS** and **CW** proved to be the most efficient. The number of the similar sentences extracted is influenced by the language and corpus type.

We consider that the established thresholds for the extraction of similar sentences suit the aim that was set. The results obtained will further be used in the template extraction process.

References

[1] D. J. Arnold, L. Balkan, S. M. R. Humphreys, and L. Sadler. *Machine Translation: an Introductory Guide*. Blackwells-NCC, London, 1994.

[2] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *Proc. of the Seventh International Symposium on String Processing and Information Retrieval - SPIRE 2000*, pages 39–48, A Curuna, Spain, September 2000. ISBN: 0-7695-0746-8.

[3] H. Camacho and A. Salhi. A string metric based on a one-to-one greedy matching algorithm. *Research in Computing Science*, 19:171–182, May 2006.

[4] I. Cicekli and A. Guvenir. *Learning Translation Templates From Bilingual Translation Examples*, volume Recent advances in Example-based Machine Translation, pages 225–286. Kluwer Acad. Publ., 2003.

[5] B. J. Dorr, P. W. Jordan, and J. W. Benoit. A survey of current paradigms in machine translation. *Advances in Computers*, 49:2–68, 1999.

[6] J. W. Hutchins. *Machine Translation: A brief History*, volume Concise History of the Language Sciences. From the Sumerians to the Cognitivists, pages 431–445. Oxford: Elsevier Science Ltd., 1995.

[7] K. McTait. *Translation Patterns, Linguistic Knowledge and Complexity in an Approach to EBMT*, volume Recent advances in Example-based Machine Translation, pages 307–338. Kluwer Acad. Publ., 2003.

[8] M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. of the international NATO symposium on Artificial and human intelligence*, pages 173–180, New York, NY, USA, 1984. Elsevier North-Holland, Inc.

[9] SimMetrics. http://www.dcs.shef.ac.uk/ sam/simmetrics.html.

[10] H. Somers. *An Overview of EBMT*, volume Recent advances in Example-based Machine Translation, pages 3–57. Kluwer Acad. Publ., 2003.

[11] C. Vertan and V. E. Martin. Experiments with matching algorithms in example based machine translation. In *In Proceedings of the International workshop "Modern approaches in Translation Technologies", in conjunction with RANLP*, September 2005.