Syntax-based Multi-system Machine Translation

Matīss Rikters[1], Inguna Skadiņa[2]
[1,2]University of Latvia, 19 Raina Blvd., Riga, Latvia
[2]Institute of Mathematics and Computer Science, 29 Raina Blvd., Riga, Latvia
E-mail: matiss@lielakeda.lv, inguna.skadina@lumii.lv

Abstract

This paper describes a hybrid machine translation system that explores a parser to acquire syntactic chunks of a source sentence, translates the chunks with multiple online machine translation (MT) system application program interfaces (APIs) and creates output by combining translated chunks to obtain the best possible translation. The selection of the best translation hypothesis is performed by calculating the perplexity for each translated chunk. The goal of this approach is to enhance the baseline multi-system hybrid translation (MHyT) system that uses only a language model to select best translation from translations obtained with different APIs and to improve overall English – Latvian machine translation quality over each of the individual MT APIs. The presented syntax-based multi-system translation (SyMHyT) system demonstrates an improvement in terms of BLEU and NIST scores compared to the baseline system. Improvements reach from 1.74 up to 2.54 BLEU points.

Keywords: hybrid machine translation, multi-system translation, syntactic parsing, under-resourced languages

1. Introduction

Multi-system machine translation (MMT) is a type of hybrid machine translation (HMT) where multiple MT systems are combined in a single system in order to boost the accuracy and fluency of the translations. It is also referred to as multi-engine MT (Mellebeek et al., 2006), coupling MT (Ahsan and Kolachina, 2010), or just MT system combination (Barrault, 2010).

Different approaches have been proposed for MMT. Traditional MSMT (Hildebrand, and Vogel, 2009) selects the best translation from a list of possible candidate translations generated by different MT engines using n-gram approach. Improvement has been reported when translated from French (+1.6 BLEU), German (+1.95 BLEU) or Hungarian (+1 BLEU) into English. However, application of similar approach for English-Latvian MT has resulted in insignificant improvement by only +0.12 BLEU points (Rikters, 2015).

Recently Freitag et al. (2015) presented a novel system combination approach that enhances the traditional confusion network system combination approach (Heafield et al., .2009) with an additional model trained by a neural network. The proposed approach yielded in translation improvement from up to +0.9 points in BLEU and -0.5 points in TER for Chinese-English and Arabic-English.

This paper presents a method that allows improving the MMT approach by incorporating syntactic information. These experiments were inspired by analysis of typical errors produced by statistical MT engines when translation is performed into a morphologically rich language with rather free order – Latvian (Skadiņa et al., 2012). This error analysis showed that the main type of errors is wrong inflection, which is usually caused by ignoring syntax rules. Our hypothesis is that translation of smaller, linguistically motivated chunks can improve this situation.

We aim to enhance the simple baseline multi-system hybrid translation (MHyT) system[1] (Rikters, 2015) and to improve MT quality for English-Latvian texts over each of the individual MT APIs.

The experiments described in this paper use multiple combinations of outputs from two or three English-Latvian MT systems. We used two MT systems – *Google Translate* and *Bing Translator* - by global developers and an English-Latvian MT system developed by Tilde company with a long-term experience in development of customized MT solutions for under-resourced languages. We believe that the syntax-based combination of two MT systems from companies that have access to enormous language resources with an MT system which is tailored for the under resourced language Latvian, allows to improve translation quality.

In the paper, we analyse combination of all three MT systems as well as combinations of system pairs. The automatic evaluation results obtained with this hybrid system are analysed and compared with human evaluation results.

The framework developed within this work allows the application of proposed strategy to other language pairs for which MT APIs are available. The developed SyMHyT framework is freely available on GitHub[2].

[1] Multi-System Hybrid Translator is available at: https://github.com/M4t1ss/Multi-System-Hybrid-Translator

[2] Syntax-based Multi-System Hybrid Translator is available at: https://github.com/M4t1ss/Multi-System-Hybrid-Translator/tree/Syntactic-Multi-System-Hybrid-Translator

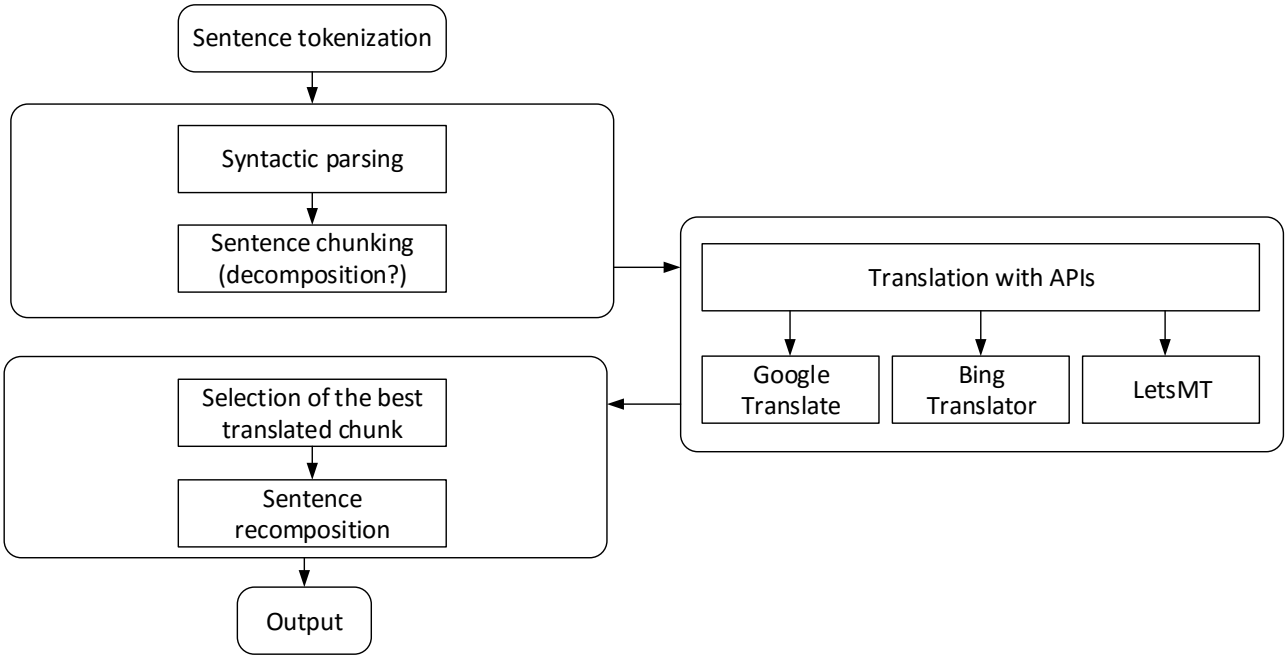

Figure 1: General workflow of the translation process

2. System's Architecture

The hybrid system described in this paper consists of three main components – 1) pre-processing of the source sentences, 2) the acquisition of a translations via online APIs and 3) post-processing - the selection of the best translation of chunks and generation of MT output. A visualized workflow of the system is presented in Figure 1.

For translation three translation APIs are used. Each translation API in our system is defined with a function that has source and target language identifiers and the source chunk as input parameter and the target chunk as the only output. This makes the system's architecture flexible allowing to integrate more translation APIs easily.

Although the system is configured to translate from English into Latvian, the source and target languages could also be changed to other language pairs that are supported by the MT APIs. Changing source language involves need for a parser that is compliant with the Berkeley Parser (Petrov et al., 2006).

2.1 Pre-processing

The aim of the pre-processing step is to divide sentences into linguistically motivated chunks that will be then translated with the on-line APIs. For this task, the Berkeley Parser is used.

The parse tree of each sentence is then processed by the chunk extractor to obtain the top-level sub-trees (noun phrases, verb phrases, prepositional phrases, etc.). This step relies only on source language parser and does not take into account properties of the target language, i.e., it is independent from the target language.

2.2 Translation with the APIs

In the scope of the paper, three online translation APIs were used – *Google Translate*[3], *Bing Translator*[4] and *LetsMT*![5]. The less known LetsMT! (Vasiljevs et al., 2012) is full-service platform that gathers public and user-provided MT training data and allows users to create custom MT systems by combining and prioritising this data. The training and translation facilities of LetsMT! are based on the open source toolkit Moses (Koehn et al., 2007). LetsMT! also provides access to a wide range of MT systems for different language pairs. These systems can be accessed using LetsMT! API for MT integration.

These specific APIs were selected because of their public availability and descriptive documentation as well as the wide range of languages that they support. One of the main criteria when searching for translation APIs was the possibility to translate from English into Latvian.

2.3 Selection of the best translated chunk

The selection of the best-translated chunk is performed by calculating the perplexity for each translation hypothesis with KenLM (Heafield, 2011). Sentence perplexity has been proven to correlate with human judgments and BLEU scores, and it is a good evaluation method for MT without reference translations (Gamon et al., 2005). It has been also used in other previous attempts of MMT to score output from different MT engines as mentioned by Callison-Burch et al. (2001) and Akiba et al. (2002). For reliable results, a large target language corpus is

[3] Google Translate API is available online at: https://cloud.google.com/translate/

[4] Bing Translator Control is available online at: http://www.bing.com/dev/en-us/translator

[5] LetsMT! Open Translation API is available online at: https://www.letsmt.eu/Integration.aspx

necessary.
When the best translation for each chunk is selected, the translation of the full sentence is generated by concatenation of chunks.

2.4 Illustration of translation process

The sentence translation using syntax-based multi-system translation approach is illustrated in Figure 2.
At first, the sentence "*3. the list referred to in paragraph 1 and all amendments thereto shall be published in the official journal of the european communities.*" is parsed with Berkeley Parser. In a next step the parsed sentence is divided into 3 chunks: "*3. the list referred to in paragraph 1 and all amendments thereto", "shall be published in the official journal of the european communities" and "*.". Each chunk is then translated with online APIs. Obtained three translations for each chunk are then evaluated and the best translation for the chunk is selected. Finally, the output is generated.

3. Experiments

This section describes the experiments performed to test the proposed syntax-based multi-system translation approach.

3.1 Data

The experiments were conducted on the English – Latvian part of the JRC-Acquis corpus version 3.0 (Steinberger et al., 2006). The corpus contains 1.4 million unique legal domain sentences. For selection of best hypothesis, a 5-gram language model was trained using KenLM.
For tests, 1581 randomly selected sentences from the JRC-Acquis corpus were used.

3.2 System combination

The proposed method was applied to all combinations of two and then all three APIs. As a result, seven different translations for each source sentence were obtained. *Google Translate* and *Bing Translator* APIs were used

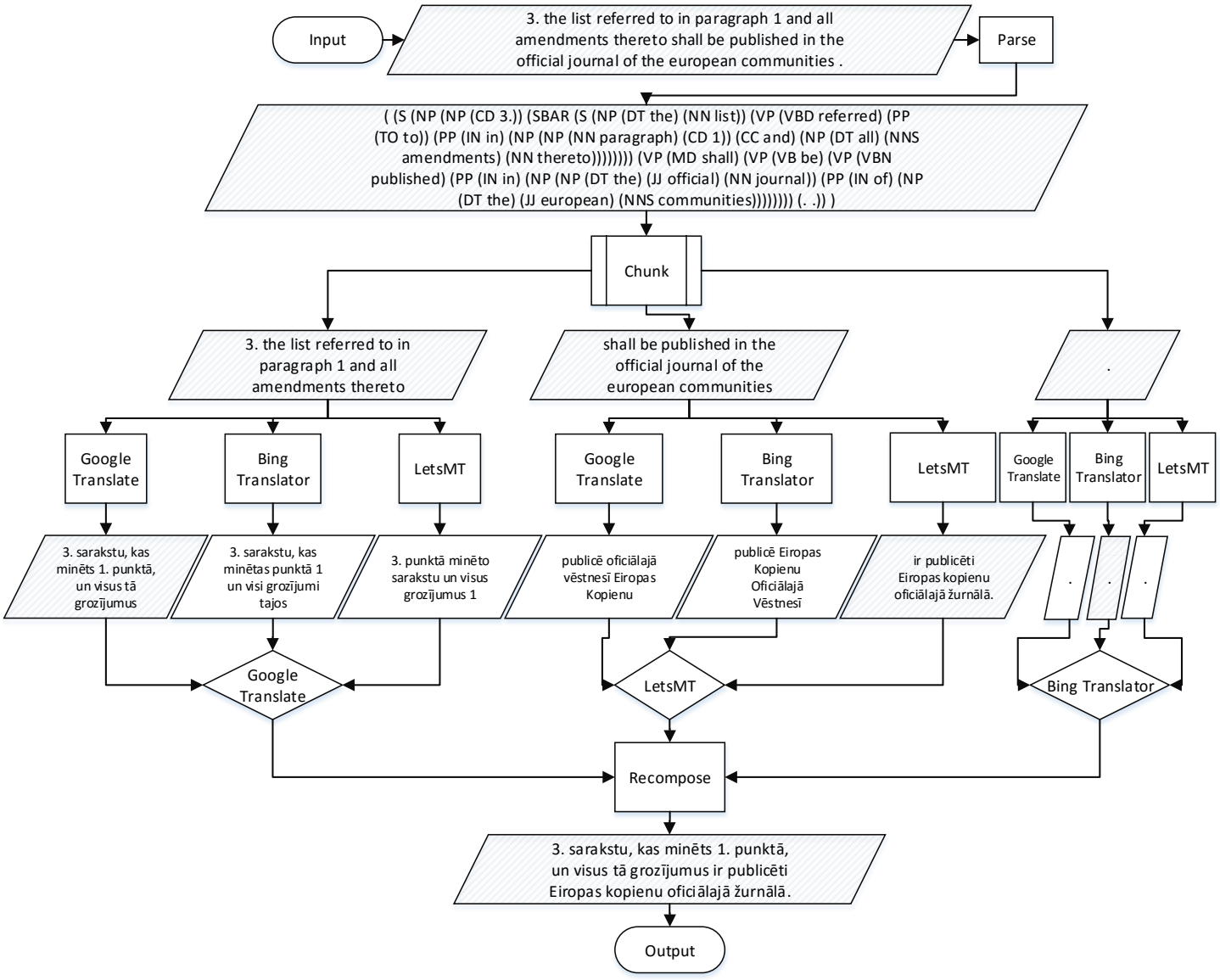

Figure 2: Illustration of the syntax-based multi-system translation approach

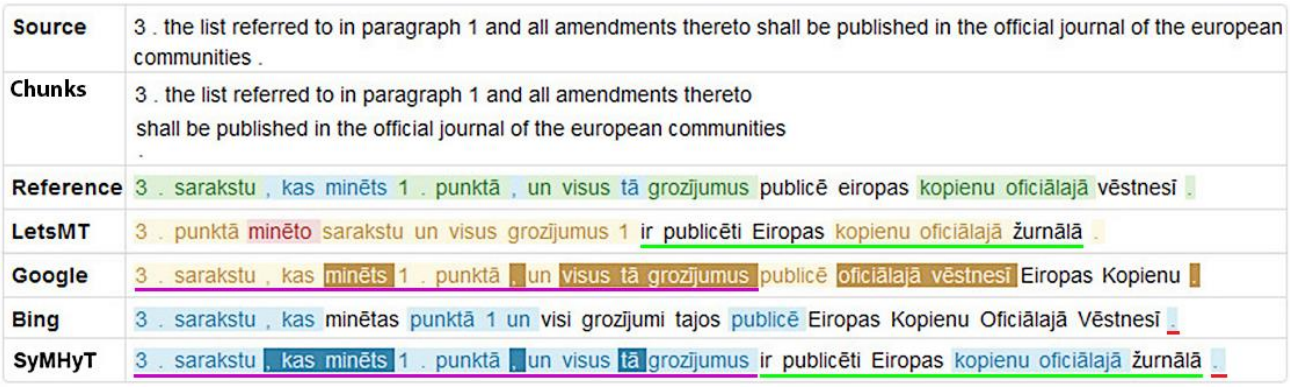

Figure 3: Comparison of translations of a sentence with the different systems with MT-ComparEval

with the default configuration and the LetsMT! API used the configuration of TB2013 EN-LV v03.

3.3 Automatic evaluation

Output of each system was evaluated with two scoring methods – BLEU (Papineni et al., 2002) and NIST (Doddington, 2002).

The results of the automatic evaluation are summarized in Table 1.

System	BLEU		NIST	
	MHyT	SyMHyT	MHyT	SyMHyT
Google Translate	18.09		8.37	
Bing Translator	18.87		8.09	
LetsMT!	**30.28**		**9.45**	
Google + Bing	18.73	21.27	7.76	8.30
Google + LetsMT	24.50	26.24	9.60	9.09
LetsMT! + Bing	24.66	26.63	9.47	8.97
Google + Bing + LetsMT!	22.69	24.72	8.57	8.24

Table 1: Evaluation results: MHyT – baseline hybrid system, SyMHyT – syntax-based hybrid system

The evaluation results clearly show an improvement over the baseline hybrid system (MHyT) that does not have a syntactic pre-processing step and thus selects the best translation from translations of full sentences.

The combination of *Google Translate* and *Bing Translator* shows about +2 BLEU improvement compared to each of the baseline systems.

Surprisingly all hybrid systems that include the LetsMT! API produce lower results than the baseline LetsMT! system.

Thus, resulting translations were inspected with the Web-based MT evaluation platform MT-ComparEval (Klejch et al., 2015) to determine, which system from the hybrid setups was selected to get the specific translation for chunk. Table 2 shows the percentage of translations from each API for the hybrid systems.

System	Google	Bing	LetsMT
Google Translate	100%	-	-
Bing Translator	-	100%	-
LetsMT	-	-	100%
Google + Bing	74%	26%	-
Google + LetsMT	25%	-	74%
LetsMT!+ Bing	-	24%	76%
Google + Bing + LetsMT	17%	18%	65%

Table 2: Distribution of selected chunks from different MT APIs

Contrary to the baseline hybrid system (Google - 28.93%, Bing - 34.31%, LetsMT! - 33.98%, equal - 2.78%) the SyMHyT system tends to use more chunks from LetsMT!. This resulted in increase of the BLEU score by +1.7 - 2.03 points over the baseline hybrid solution.

Figure 3 shows an example of the source sentence, extracted chunks, reference sentence, and all system translations, including the hybrid SyMHyT, with the differences highlighted. The purple line highlights the chunk selected from *Google Translate*, the red line – the chunk from *Bing translator*, and the green line – the chunk from LetsMT!. It can be seen that the hybrid system (SyMHyT) used the first chunk from *Google's* output and the second chunk from LetsMT.

This illustration also shows weakness of the proposed approach – selected chunks are very long and are independent from the target language. Our hypothesis is that this is the reason why the hybrid approach did not perform better as LetsMT! system.

3.4 Experiments with different language models

To evaluate the influence of language model size on the chunk selection process we trained two 12-gram language models – one on the same JRC-Acquis corpus and another one on the DGT-TM corpus (Steinberger et al., 2013). The results of this experiment are presented in Table 3.

LM	Size (sentences)	BLEU
5-gram JRC	1.4 million	**24.72**
12-gram JRC	1.4 million	24.70
12-gram DGT	3.1 million	24.04

Table 3: Influence of different language models

For this approach the higher order language model did not show improvement. Some additional experiments described in Rikters and Skadiņa (2016) using 6-gram, 9-gram and 12-gram LMs resulted in slightly higher BLEU score but the change was not statistically significant.

3.5 Application of random chunks

To justify that our approach that uses the linguistically motivated chunks are much better as just cutting sentences into random chunks we performed three experiments. The sentence was split into 5-grams in one experiment (+ one shorter n-gram, if the last one is made up of less tokens), random 1-grams to 4-grams in the second experiment and random 1-grams to 6-grams in the last experiment. We used the same 5-gram JRC-Acquis language model for best translation selection. Results of these experiments (Table 4) fully confirmed our hypothesis of advantage of linguistically motivated chunks.

Chunks	BLEU
SyMHyT chunks	**24.72**
5-grams	11.85
Random 1-4 grams	7.33
Random 1-6 grams	10.25

Table 4: Influence of different chunk selection strategies on MT output

4. Human Evaluation

A random 2% (32 sentences) of the translations from the experiment were given to 10 native Latvian speakers with instructions to evaluate fluency and adequacy. The MT-EQuAl tool (Girardi et al., 2014) was used for evaluation. The three baseline systems were compared with the syntax-based hybrid system that combines all three baselines. Evaluators were instructed to mark each sentence with one of the following labels: “most fluent translation”, “most precise translation”, “neither most fluent, nor most precise”, or “both most fluent and most precise”. In case, if a translation is marked as most fluent and adequate, then all others alternatives needed to be marked as “neither most fluent, nor most precise”.

The results of evaluation are summarized in Table 5. The free-marginal kappa (Randolph, 2005) for these annotations is 0.335 that indicates substantial agreement between the annotators.

System	Fluency AVG	Accuracy AVG	SyMHyT selection	BLEU
Google	35.29%	34.93%	16.83%	18.09
Bing	23.53%	23.97%	17.94%	18.87
LetsMT	20.00%	21.92%	65.23%	30.28
SyMHyT	21.18%	19.18%	-	24.72

Table 5: Manual evaluation results

As it can be seen from the table, about 1/3 of translations recognized by annotators as most fluent and most adequate are translations from *Google Translate* system. This contradicts with the automatic evaluation results and the selections made by the syntax-based hybrid MT, where a tendency towards the LetsMT! translation is observed.

Inspecting the annotations closer, we performed a broader analysis of this result. Our hypothesis is that LetsMT! was chosen less often by the annotators because of failure to translate dates or numbers in specific sentences while the rest of the sentence was very similar to the reference, hence scoring more BLEU points. Closer inspection revealed that three sentences from LetsMT! contained “βNUMβ” tag, which appears to be an error in the named entity processor during time of experiments. There were also five sentences that contained untranslated dates, e.g., “31 december 1992” or “february 1995.” These errors account for LetsMT! not be selected by annotators in 25% cases of the evaluation dataset, while in case of BLEU score, their influence was not so significant.

5. Conclusion

This paper described an improved machine translation system combination approach for public online MT system APIs that uses syntactic and statistical features. All test cases showed an improvement in BLEU score when compared to the baseline hybrid system and improvement in NIST score in one case. When used only with *Google Translate* and *Bing Translator,* the SyMHyT approach resulted in +2.4 BLEU points compared to the best individual API.

For hybrid systems that included the LetsMT! API a decrease in BLEU was observed. This can be explained by the scale of the engines - the *Bing* and *Google* systems are more general, designed for many language pairs, whereas the MT system in LetsMT! is customized for English – Latvian translations.

The proposed method for chunking is very straightforward and easily accomplishable. In later experiments (Rikters and Skadiņa, 2016), we used a more sophisticated chunker that is slightly more dependent on the source language, as it includes additional rules for chunk selection.

The described system is in the second phase of its lifecycle and further enhancements are planned. Several methods could improve the current system combination approach. Improvements are planned for both - the chunk selection step and the selection of the best-translated

chunk.

In the presented approach, the chunker splits sentences in top-level chunks without analysis of sub-chunks or cases when a chunk is single token. However, the larger chunks should be split in smaller sub-chunks and the single-word chunks should be combined with the neighbouring longer chunks. The better results could be achieved if sentence is divided into certain types of phrases, e.g. noun phrases and verb phrases, but not prepositional phrases, infinitive phrases, etc. Another approach would be to introduce language pair specific constituents, as it has been done by Marton et al (2012) in Hiero framework.

There are also several possible areas of improvement for the selection of the best translation, for instance, usage of confusion networks, neural network language models or a language model of morpho-syntactic tags.

6. Acknowledgements

The research was supported by Grant 271/2012 from the Latvian Council of Science. The authors would like to thank Mārcis Pinnis for providing tools for kappa calculations, as well as comments and suggestions. We would also like to thank reviewers for their valuable comments and suggestions for further work.

7. Bibliographical References

Ahsan, A., Kolachina, P. (2010). Coupling Statistical Machine Translation with Rule-based Transfer and Generation. In *AMTA-The Ninth Conference of the Association for Machine Translation in the Americas*. Denver, Colorado.

Akiba, Y., Watanabe, T., and Sumita, E. (2002). Using language and translation models to select the best among outputs from multiple MT systems. Proceedings of the *19th international conference on Computational linguistics*-Volume 1. Association for Computational Linguistics.

Barrault, L. (2010). MANY: Open source machine translation system combination. *The Prague Bulletin of Mathematical Linguistics* 93: 147-155.

Callison-Burch, C., Flournoy, R. S. (2001). A program for automatically selecting the best output from multiple machine translation engines. Proceedings of the *Machine Translation Summit VIII*.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. Proceedings of *the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc.

Freitag, M., Peter, J., Peitz, S., Feng, M., Ney, H. (2015). Local System Voting Feature for Mac-hine Translation System Combination. In *EMNLP 2015 Tenth Workshop on Statistical Machine Translation* (WMT 2015), pages 467-–476, Lisbon, Portugal.

Gamon, M., Aue, A., Smets, M. (2005). Sentence-level MT evaluation without reference translations: Beyond language modeling. Proceedings of *EAMT*.

Girardi, C., Bentivogli, L., Farajian, M. A., Federico, M. (2014). MTEQuAl: a Toolkit for Human Assessment of Machine Translation Output. In *COLING 2014*, pages 120–123, Dublin, Ireland, Dublin City University and ACL.

Heafield, K., Hanneman, Gr., Lavie, A. (2009). Machine Translation System Combination with Flexible Word Ordering. Proc *4th Workshop on SMT*, Athens

Heafield, K. (2011). KenLM: Faster and smaller language model queries. Proceedings of *the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2011.

Hildebrand, A. S., Vogel, St. (2009). CMU System Combination for WMT'09. Proc 4th Workshop on SMT, Athens

Yuval, M., Chiang, D., Resnik, P. (2012). Soft syntactic constraints for Arabic-English hierarchical phrase-based translation. *Machine Translation* 26(1–2):137–157.

Klejch, O., Avramidis, E., Burchardt, A., Popel, M. (2015). MT-ComparEval: Graphical evaluation interface for Machine Translation development. *The Prague Bulletin of Mathematical Linguistics* 104.1: 63-74.

Koehn, P., Federico, M., Cowan, B., Zens, R., Duer, C., Bojar, O., Constantin, A., Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. Proceedings of the *ACL 2007* Demo and Poster Sessions, 177-180. Prague.

Mellebeek, B., Owczarzak, K., Van Genabith, J., Way, A. (2006). Multi-engine machine translation by recursive sentence decomposition. Proceedings of *the 7th Conference of the Association for Machine Translation in the Americas*, 110-118.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J. (2002). BLEU: a method for automatic evaluation of machine translation. Proceedings of *the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics.

Petrov, S., Barrett, L., Thibaux, R., Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. Proceedings of *the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Skadiņa I., Levāne-Petrova, K., Rābante, G. (2012). Linguistically Motivated Evaluation of English-Latvian Statistical Machine Translation. // Human Language Technologies – The Baltic Perspective - Proceedings of *the Fifth International Conference Baltic HLT 2012*, IOS Press, Frontiers in Artificial Intelligence and Applications, Vol. 247, pp. 221-229.

Randolph, J. J. (2005). Free-Marginal Multirater Kappa (multirater K [free]): An Alternative to Fleiss' Fixed-Marginal Multirater Kappa. Online Submission.

Rikters, M. (2015). Multi-system machine translation using online APIs for English-Latvian. *ACL-IJCNLP 2015*: 6.

Rikters, M., Skadiņa, I. (2016). Combining machine translated sentence chunks from multiple MT systems.

CICLing. 2016.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv* preprint cs/0609058.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S. Schlüter, P. (2013). "Dgt-tm: A freely available translation memory in 22 languages." *arXiv* preprint arXiv: 1309.5226 (2013).

Vasiļjevs, A., Skadiņš, R., Tiedemann, J. (2012). LetsMT!: A Cloud-Based Platform for Do-It-Yourself Machine Translation. Proceedings of the *ACL 2012 System Demonstrations*, 43–48, Jeju Island, Korea: Association for Computational Linguistics.