Named Entities from Wikipedia for Machine Translation*

Ondřej Hálek, Rudolf Rosa, Aleš Tamchyna, and Ondřej Bojar
ohalek@centrum.cz, rur@seznam.cz, a.tamchyna@gmail.com, bojar@ufal.mff.cuni.cz

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Abstract. *In this paper we present our attempt to improve machine translation of named entities by using Wikipedia. We recognize named entities based on categories of English Wikipedia articles, extract their potential translations from corresponding Czech articles and incorporate them into a statistical machine translation system as translation options. Our results show a decrease of translation quality in terms of automatic metrics but positive results from human annotators. We conclude that this approach can lead to many errors in translation and therefore should always be combined with the standard statistical translation model and weighted appropriately.*

1 Introduction

Translation of named entities (NE) is an often overlooked problem of today's machine translation (MT). Particularly, most statistical systems do not handle named entities explicitly, simply relying on the model to pick the correct translation. Since most of NEs are rare in texts, statistical MT systems are incapable of producing reliable translations of them.

Moreover, many NEs are composed of ordinary words, such as the term "Rice University". In the attempt to output the most likely translation, a statistical system would translate this collocation word by word.

In this paper, we attempt to address this problem by using Wikipedia[1] to translate NEs and present them already translated to the MT system.

1.1 Named Entity Translation Task

The set of named entities is unbounded and there are many definitions of named entities. In our project, we work with a vague definition of a *named entity being a word or group of words which, when left untranslated, are a valid translation anyway* (despite the fact that a "real" translation is usually better if it exists; however, it does not exist in many cases).

Translation of named entities consists of several subtasks. NEs have to be identified in the source text and their translations must be proposed. These have to be appropriately incorporated into the sentence translation — the sentence context must match the NE and vice versa.

For the English-Czech language pair, matching NEs to the sentence context consists mainly of inflection of NE words. For example, while "London" translates to Czech as "Londýn", in the context of a more comlex NE, the name has to be inflected in Czech, such as "London airport" → "Londýn**ské** letiště" ($\text{London}_{\text{adj}}$ airport).

Matching the sentence context to the named entity is needed when some information, such as the grammatical gender, comes from the NE. For example, Czech verbs in past tense have different forms for each gender — the verb "came" has to be translated as "přišel" when the subject is masculine, as "přišla" for feminine and as "přišlo" for neuter subject. This information needs to be taken into account in translation: "Jeffry came." → "Jeffry **přišel**.".

1.2 Work Outline

We experiment with English to Czech translation.

Named entity recognition is done in two steps. First, all potential NEs are recognized using a simple recognizer with a low precision but with a high recall. Then, confirmation/rejection of named entities is done — if there is an article with the corresponding title in English Wikipedia, we try to confirm the potential NE as a true NE based on the categories of the article.

The translation of a NE is done by looking up the Czech version of the English Wikipedia article about the named entity. Its title is considered the "base translation". Other potential translations (in our case this means simply various inflected forms) are then extracted from the text of the Czech article. Each named entity found in the input text is then replaced with a set of its potential translations, from which the MT system then tries to choose the best one.

The matching of the sentence context to the NE is not handled explicitly. We rely on target-side language model to determine the most appropriate option.

* This work has been supported by the grants EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic), P406/11/1499, and MSM 0021620838.

[1] http://en.wikipedia.org/

2 Recognition of Potential Named Entities

In our case, the goal of potential NE recognition is to find as many potential NEs as possible (i.e. we favour higher recall at the expense of precision), because the candidates for NEs are still to be confirmed or rejected in the next step. Thanks to the external world knowledge provided by Wikipedia, our task is not a typical NER scenario. NE recognition is not the focal point of our experiment, so we limit ourselves to using two tools for recognition of potential NEs: our simple named entity recognizer and Stanford named entity recognizer.

2.1 Simple Named Entity Recognizer

We created a simple rule-based named entity recognizer for selecting phrases suspected to be named entities. It looks for capitalized words and uses a small set of simple rules for beginnings of sentences — most notably, the first word of a sentence is a potential NE if the following word is capitalized (except for words on a stoplist, such as "A", "From", "To"...). Sequences of potential NEs are always considered as a single one multiword potential NE.

2.2 Stanford Named Entity Recognizer

The Stanford NER [4] is a well-known tool with documented accuracy over 90% when analyzing named entities according to CoNLL Shared Task [12]. However, this classification does not match our named entity definition, and we also use only a limited recognition model.[2]

2.3 Evaluation of Named Entity Recognizers

To evaluate the tools we use an evaluation text consisting of 255 sentences rich in named entities, originally collected for a quiz-based evaluation task [1]. The sentences are quite evenly distributed among four topics — directions, meetings, news and quizes.

We first performed a human annotation of NEs in the evaluation text, where two annotators marked NEs in the text according to our NE definition. The inter-annotator agreement F-measure[3] was only 83%, which sets an upper bound on the value for our automatic recognizers. We then picked one annotation as a standard, according to which we compare outputs of the NE recognition tools.

To measure the **precision** of a NE recognizer, we count the NEs on which the tool agrees with the standard annotation and divide it by the total number of NEs recognized by the tool. Similarly, the **recall** is measured as the number of NEs confirmed by the standard divided by the number of NEs in the standard.

The performance of the two aforementioned tools measured on the evaluation text is shown in Table 1.

Table 1. Comparison of NE recognizers.

Recognizer	Precision	Recall	F-measure
Simple NER	0.57	**0.73**	**0.64**
Stanford NER	**0.70**	0.49	0.58

Our Simple NER has a significantly higher recall than Stanford NER; it is actually capable of delivering most of the named entities. Its low precision is not an issue for our experiment since in the next step we confirm the named entities by using Wikipedia categories. Its F-measure is also higher than that of Stanford NER, suggesting the Simple NER suits our NE definition better.

Since the Stanford NER results are well documented, we assume that its poor results in our experiment are mainly caused by a different NE definition and the recognition model used — in this setup Stanford NER recognizes only people, locations and organizations, but e.g. named entities from the software class (names of programs, programming language functions etc.) are left out from the recognition.

On the other hand, with Stanford NER we are capable of correctly recognizing complex named entities, and the recall of recognition of named entities at sentence beginnings is higher than that of Simple NER.

3 Confirmation of NEs by Wikipedia

For each potential named entity we try to confirm it as a true named entity using Wikipedia categories.

First we look for the article on English Wikipedia with a title matching the potential NE. If it does not exist, we reject it immediately.

We then get the categories of that article. For each category we do a search for its superior categories (several hard limits had to be introduced, because the categories do not form a tree, not even a DAG; the maximum depth of the search was set to 6).

In the end, the categories found are compared with our hand-made list of named entity categories. If at

[2] `ner-eng-ie.crf-3-all2008-distsim` — a conditional random field model that recognizes 3 NE classes (Location, Person, Organization) trained on unrestricted data, uses distributional similarity features

[3] $F = \frac{2PR}{P+R}$, where P stands for precision and R for recall

```
http://en.Wikipedia.org/w/api.php?action=query&prop=categories&redirects&clshow=!hidden
&format=xml&titles=Rice_University

<?xml version="1.0"?>
  <api>
    <query>
      <pages>
        <page pageid="25813" ns="0" title="Rice University">
          <categories>
            <cl ns="14" title="Category:Association of American Universities" />
            <cl ns="14" title="Category:Educational institutions established in 1891" />
            . . .
```

Fig. 1. Example of XML Response to a Request to Wikimedia API

least one of the article categories or their super-categories is contained in the NE categories list, we confirm the potential NE as a true NE; otherwise we reject it.

The following categories are considered to indicate NEs:

- Places
- People
- Organizations
- Companies
- Software
- Transport Infrastructure

To get the information from Wikipedia we use the Wikimedia API [7]. Figure 1 shows an example of the API response.

4 Wikipedia Translation

For each English Wikipedia article about a NE we look if there is a corresponding Czech article (this is provided by Wikipedia under the page section "Languages"). If there is one, we use its title as the base translation.

We then try to find all inflected forms of the base translation in the text of the Czech article to use as alternative translations.

For each word in the base translation, we trim its last three letters, keeping at least the first three letters intact. This is considered a "stem".

Then, the Czech article is fetched using Wikimedia API and wiki markup is stripped. We then search the article text for sequences of words with the same stems. If we find a match, we consider it an inflected form of our base translation and include it in the list of potential translations.

Finally, we estimate the probability of the various forms from their counts of occurrences.

5 Translation Process

In order to utilize the retrieved translation suggestions, we had to find a way of incorporating them as additional translation options for the decoder. This can be generally done in several ways, such as by extending the parallel data, by adding new entries into the translation model (i.e. the phrase table), or by preprocessing the input data.

We use the Moses [6] decoder throughout our experiments. Input pre-processing can be realized fairly easily in Moses via XML markup of the input sentences. It is simple to incorporate alternative translations for sequences of words and even to assign the translation probability for each of the options. The markup of input data is illustrated in Figure 2.

When scoring hypotheses, Moses uses several translation model scores, namely $p(e|f)$, $p(f|e)$, $lex(e|f)$ and $lex(f|e)$, i.e. translation probabilities in both directions (where f stands for "foreign" (English in this case) and e stands for Czech) and lexical weights. The value specified in the markup (or 1 if omitted) replaces *all* of these scores.

Pre-processing of the input data also has the advantage of not requiring to retrain or modify existing translation models. Fully trained MT systems can therefore be easily extended to take advantage of our method.

Moses can treat the translation suggestions as either *exclusive* or *inclusive*. If set to *exclusive*, only options suggested in the input markup are considered as translation candidates. With the *inclusive* setting, these options are included among the suggestions from the translation model, competing with them for the highest score. Depending on the quality of the translation model and the external translation suggestions, this setting can either improve or hurt translation performance.

When estimating the probability of our translations, we distribute the *whole* probability mass among

They moved to <name translation="Londýn||Londýna" probs="0.6||0.4">London</name> last year.

Fig. 2. An example of including external translation options using XML markup of input.

them. The scores of translation suggestions provided by the translation model are typically much lower. However, target language model usually has a significant impact on hypothesis scoring, so even if the external translation scores are set to unrealistically high values, the language model makes the "competition" with translation model options reasonably fair.

The default settings for common language models, such as SRILM or KenLM, as used in Moses, assign *zero* log-probability (i.e. the probability of 1) to unknown tokens instead of the intuitive $-\infty$. In most cases, training data of the language model for the target language also include the target language part of the translation model parallel data, so this is not an issue. However, our translation suggestions often contain tokens unseen in any data, including some noise introduced by the imperfect suffix trimming heuristic. Instead of penalizing such options, the language model *promoted* them, since the unknown words were ignored and therefore did not lower the overall ngram probability (any known token has a probability < 1, scoring inevitably lower). We were able to solve this problem by setting a very low probability for unknown tokens. Perhaps a more interesting option would be to add the full texts of the Czech Wikipedia articles to the language model. This would ensure the translation of the NE is known to the language model and even including some plausible contexts. We leave this for future research.

6 Experimental Results

We conducted a series of translation experiments, evaluating various setups of our method. We also carried out a blind manual evaluation, in which the annotators compared outputs of two MT setups which used our method and of the baseline MT system.

6.1 Data Sources

We used CzEng 0.9 [2] as the source of both parallel and monolingual data to train our MT system. CzEng is a parallel richly annotated Czech-English corpus. It contains roughly 8 million parallel sentences from a variety of domains, including European regulations (about 34% of tokens), fiction (15%), news (3%), technical texts (10%) and unofficial movie subtitles (27%). In all our experiments we used 200 thousand parallel sentences for the translation model and 5 million monolingual sentences for the target language model. We also used CzEng as a source of a separate set of 1000 sentences for tuning the model weights and another 1000 sentences for automatic evaluation.

Since manual evaluation would benefit from data rich in terms of named entity occurrences, we used the same set of sentences as in NER evaluation. These sentences cover quite a wide range of topics, so they seem suitable even for translation evaluation.

6.2 Tools

We used the common pipeline of popular tools for phrase-based statistical MT, namely the Moses decoder and toolkit, SRILM language modelling tool [11], an open-source implementation of IBM models GIZA++ [8] for obtaining word alignments. KenLM [5] was used instead of SRILM during decoding for its better speed and simplicity.

We used the MERT (Minimum Error Rate Training) [9] algorithm to tune weights of the log-linear model and BLEU [10] as the de-facto standard automatic translation quality metric.

6.3 Automatic Evaluation

We evaluated a small subset of possible setups, all our results are summarized in Table 2. The main goal of these experiments was to determine which components of our pipeline are actually important for achieving good results.

We began with a simple scenario, only using the titles of the articles for translation (i.e. inflected occurrences of the title were not available to the decoder) and forcing Moses to use only our suggestions when translating a NE in a sentence.

In the very first case, we also kept *unknown* named entities in their original form — by an *unknown* NE we understand an entity for which the corresponding English Wikipedia article exists and its categories imply that it is a named entity, but there is no corresponding Czech article. Since the Czech version of Wikipedia is much smaller, this case occurs quite often.

The BLEU score in these simple scenarios confirms our expectations — in statistical machine translation, forcing or limiting translation possibilities rarely helps. More specifically, by excluding phrase table entries, we forbid the log-linear model to use potentially more adequate translations. The phrase table may well include

Table 2. BLEU scores of our setups and the baseline system.

NEs Suggested	Regular Translations	Unknown NEs	NER	BLEU
Only base forms	Excluded	Preserved	Simple	25.13
Only base forms	Excluded	Translated	Simple	25.38
Only base forms	Included	Translated	Simple	25.80
All forms	Included	Translated	Simple	25.97
All forms	Included	Translated	Stanford	25.98
Baseline				**26.62**

many variants of a given named entity translation, providing more context and inherent disambiguation. This information should be used and possibly even preferred to a single translation or an enumeration of potential translations suggested by our tools (albeit probabilistically weighted). On the other hand, promoting phrase table entries too eagerly would result in undesirable translations in some cases, for example when a named entity is composed of common words.

It is also not surprising that keeping *unknown* entities untranslated hurts (automatically estimated) translation performance, as Czech tends to translate most of frequent foreign names, and even NEs which are used in their original form are usually inflected in Czech. NEs that would remain completely unchanged are quite rare. Sentences with some NEs left untranslated may be more understandable, even considered better translations in some cases, but BLEU score is necessarily worse.

When we allowed translation model entries to compete with our suggestions, the score improved further to 25.80. The target language model was apparently able to promote options from the phrase table in spite of their low translation model scores compared to our suggestions (see Section 5).

Our translations could have been inadequate for two main reasons in this scenario:

- Lexically incorrect translation,
- Wrong surface form (only title translation used).

Adding a full list of all inflected forms of NEs along with their estimated probabilities improved the translation quality slightly, presumably because the target language model was able to determine which of our suggestions fitted best into the sentence translation.

We can therefore conclude that our approach to incorporating named entity translations works successfully — the outputs contained some direct translations of article titles, some inflected forms extracted from the article content and some phrase table entries.

Using Stanford named entity recognizer brought no further gains. The recognizer marked a different (albeit smaller) set of NEs, but further filtering based on Wikipedia article categories and the absence of many Czech equivalent articles made the difference negligible.

Finally, all our scenarios scored worse than the baseline in terms of BLEU. While we believe that the motivation behind our method is valid, we were not able to avoid some errors in each of the steps that, when combined, resulted in a loss in BLEU score. A detailed analysis of errors is provided in Section 6.5.

On the other hand, we also achieved several notable improvements in translation quality even in the CzEng test set, some of which are shown in Figure 3.

6.4 Manual Evaluation

We had four annotators evaluate 255 sentences rich in named entities, using QuickJudge[4] which randomized the input. In the input sentences there were approximately 400 named entities, but the translations differed only in 78 sentences. QuickJudge automatically skips sentences with identical translations, so the annotators only saw these 78 sentences.

Three setups were evaluated: the "Baseline" unmodified Moses system, and two modifications of that system, "Translate" and "Keep unknown". The system marked as "Translate" corresponds to the best-performing setup, not using Stanford NER. "Keep unknown" is the same system, however, unknown NEs are handled differently — if a potential NE is confirmed by Wikipedia, but a Czech translation does not exist, it is kept untranslated in the output.

The annotators were presented with the source English sentence and with three translations coming from the three different setups. Then they assigned marks 1, 2 and 3 to them. Ties were allowed and only relative ranking, i.e. not the absolute values, was considered significant.

Table 3 summarizes the results. The values suggest a large number of ties — this is not surprising since differences between systems were small, their outputs often differed only in 1 word or inflection of a named entity.

We find it promising that our setups won according to all annotators. The inter-annotator agreement

[4] `http://ufal.mff.cuni.cz/euromatrix/quickjudge/`

Source	It was **Nova Scotia** on Wednesday.	
Baseline	byl$_{masc}$ to **nova scotia** ve středu.	(NE is left untranslated)
Our setup	to bylo$_{neut}$ **nové skotsko**$_{neut}$ ve středu.	(correct NE translation and gender agreement)
Source	In August, 1860, they returned to the **Victoria Falls**.	
Baseline	v srpnu, 1860, se k vyjádření **falls**.	("Victoria" is left out, "falls" kept untranslated)
Our setup	v srpnu, 1860, se na **viktoriiny vodopády**.	(correct translation extracted from Wikipedia)

Fig. 3. Examples of translation improvements. "Our setup" denotes the best-performing setup in terms of BLEU.

was however surprisingly low — even though in total, the annotators' preferences match, the individual sentences that contributed to the results differ greatly among them. All annotators agreed on a winner in only 25% sentences.

Confirming our intuition, annotators usually preferred to keep unknown entities untranslated. The fact that all of the annotators speak English certainly contributed to this result, however we believe that keeping unknown NEs in the original form is often the best solution, especially in terms of preserved information. Imagine a translation of a guidebook, for example — if an MT system correctly detects NEs and keeps unknown ones untranslated, the result is probably better than if it attempts to translate them. Thanks to the NER enhanced by Wikipedia, our system would produce more informative translations than a standard SMT system, which tends to translate NEs in various undecipherable ways.

Table 3. Number of wins (manual annotation)

Annotator	Baseline	Translate	Keep unknown
1	46	**56**	51
2	38	45	**54**
3	41	39	**47**
4	35	43	**49**

6.5 Sources of Errors

In order to explain the drop of BLEU in a more detailed fashion, we examined the translation outputs and attempted to analyze the most common errors made by our best-performing setup.

Incorrect Wikipedia Translation Quite often, the Wikipedia article contained information about a different meaning of the term. When translated to Czech, the difference in the meaning became apparent. For example, the default Wikipedia article on "Brussels" discusses the whole "Brussels Region", therefore the Czech translation is "Bruselský region". This word appeared several times in the test data and the default interpretation was wrong in all cases.[5]

Suffix Trimming Error Suffix trimming also occasionally matched words or word sequences completely unrelated to the article name. As an example, the name of the company Nestlé matched the word "nesprávně" ("incorrectly") in the Czech article. Because this word is quite common, the language model score ensured it to appear in the final translation. A similar example was matching "pole" ("field") in the article about Poland ("Polsko" in Czech). We decided to match case-insensitively in order to cover cases of named entities that do not begin with a capital letter in Czech (such as "Gulf War", "válka v Zálivu").

Wrong Named Entity Form There are two possible causes for an error of this kind — either the Czech article did not contain the inflected form needed in the translation, or the language model failed to enforce the correct option, mainly because the NE contained words unknown to the model (never seen in the monolingual training data).

Since BLEU does not differentiate between a wrong word suffix and a completely incorrect word translation, these errors are equally severe in terms of automatic evaluation.[6] On the other hand, human annotators consider a mis-inflected (otherwise correct) translation to be better than a completely untranslated named entity.

[5] It is however noteworthy that the inflected form of this particular name was always chosen correctly.

[6] Metrics with paraphrasing (e.g. Meteor [3]) could solve a part of the issue. Another option is to replace all words with their lemmas in the hypothesis and the reference and use a standard n-gram metric like BLEU. This would completely ignore errors in word forms, which is inadequate as well and might seem manipulated.

Table 4. BLEU scores of two setups using alternative translation table and the baseline system.

NEs Suggested	Regular Translations	Unknown NEs	NER	BLEU
All forms (old)	Included	Translated	Simple	**27.11**
All forms (new)	Included	Translated	Simple	26.60
Baseline				26.62

7 Wikipedia Translations as a Separate Phrase Table

In order to incorporate weighting of our translations into MERT, we also used a contrastive setup with an alternative phrase table instead of the XML markup of input sentences. The decoder was then working with two translation tables — the standard one, generated by GIZA++ from the parallel corpus, and the new one, created by our tools. As is shown in Figure 4, there are two scores in our table — the first one is the probability assigned by our tools (based on number of occurrences of the form in the text of the Czech Wikipedia article) and the second one is the "penalty" for using our NE translation.[7] It is up to MERT to estimate the weight to assign to our translations.

```
London ||| Londýn ||| 0.4 2.718
London ||| Londýna ||| 0.2 2.718
```

Fig. 4. Example of phrase table entries.

7.1 Results

Although the results of this experiment look promising, they have not been fully evaluated yet and are therefore only preliminary. There is an improvement in BLEU score (see Table 4), but it is not a result of better NE translation. The unstability of MERT process results in different weights in both translations, causing the baseline translation and our experiment outputs to differ significantly in whole sentences, not only in NE translation. Futher analysis and experiments are therefore needed.

There are two results reported in Table 4 because two different versions of the inflector were used to get the inflected forms. The "old" one uses all text data from the body of the article (including e.g. external links), while the "new" one looks for the inflected form only in the text of the article.

[7] This penalty is used in all Moses phrase tables; it is the same for all entries and equals $2.718 \doteq \exp(1) = e$.

8 Conclusion

Our approach of automatically suggesting translations of named entities based on Wikipedia texts leads to drop in automatic evaluation but to a slight improvement in manual evaluation of MT quality. Part of this improvement is due to *not translating* identified entities at all.

While some deficiencies of the proposed method of NE translation can be hopefully mitigated (poor suffix trimming and search for various forms of target-side NEs), the incorrectness of some Wikipedia translations is not easy to solve. It is therefore questionable whether the named entity translations provided by our system should be used for all named entities, or only for entities not present (or very rare) in the training data.

We described two methods of mixing the newly proposed translations and the default translations of the MT system. We studied the XML-input method more and learned that it faces an imbalance in scoring of hypotheses from the two sources. We also report preliminary results of the other method: alternative decoding paths, allowing the model to choose the best balance automatically. While the automatic scores for the second method increased slightly, the results are not yet stable and a further analysis is needed.

In sum, we have shown that Wikipedia can serve as a valuable source of bilingual information and there is an open space for incorporating this information into machine translation. However, Wikipedia should not serve as the only source of information, and the extracted information should be confirmed e.g. by analysis of some other monolingual data.

References

1. Jan Berka, Martin Černý, and Ondřej Bojar. Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, (95):77–86, April 2011.
2. Ondřej Bojar and Zdeněk Žabokrtský. CzEng 0.9: Large Parallel Treebank with Rich Annotation. *Prague Bulletin of Mathematical Linguistics*, 92:63–83, 2009.
3. Michael Denkowski and Alon Lavie. METEOR-NEXT and the METEOR Paraphrase Tables: Improved Evaluation Support For Five Target Languages. In *Proceed-*

ings of the ACL 2010 Joint Workshop on Statistical Machine Translation and Metrics MATR, 2010.

4. Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*. The Association for Computer Linguistics, 2005.
5. Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, July 2011. Association for Computational Linguistics.
6. Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL*. The Association for Computer Linguistics, 2007.
7. MediaWiki. Mediawiki — mediawiki, the free wiki engine, 2007. [Online; accessed 23-May-2011].
8. F. J. Och and H. Ney. Improved statistical alignment models. pages 440–447, Hongkong, China, October 2000.
9. Franz Josef Och. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167, 2003.
10. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.
11. Andreas Stolcke. Srilm — an extensible language modeling toolkit, June 06 2002.
12. Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.