Translating Questions for Cross-Lingual QA

Jörg Tiedemann
Information Science
University of Groningen
PO Box 716
9700 AS Groningen, The Netherlands
j.tiedemann@rug.nl

Abstract

In this paper we investigate possibilities of the development of a task-specific translation component for cross-language question answering. We focus on the optimization of phrase-based SMT for models trained on very limited data resources. We also look at the combination of such systems with another approach based on example-based MT with proportional analogies. In our experiments we could improve a strong baseline of a general purpose MT engine with more than 5 BLEU points.

1 Introduction

Question answering (QA) is a popular task in the NLP research community. It combines various exciting sub-tasks coming from research in information extraction, retrieval and different kinds of linguistic processing in a real-world application. Cross-lingual QA adds yet another component to such systems, namely a translation component, in order to open QA systems for different languages. The motivation is to enable users to post questions in their favorite language and to make it possible to find answers in documents using other languages. So far, cross-lingual QA is mainly of academic interest because of the general shortcomings in the accuracy of QA systems and also the quality of current general-purpose machine translation (MT) engines. In this paper, we investigate the use of standard techniques in statistical MT (SMT) for the development of a task-specific translation component to be integrated in a cross-lingual QA application. However, we focus exclusively on this component without evaluating the effect of translation quality on a particular QA engine.

The predominant approach to cross-lingual QA is to translate incoming questions into the language the QA system understands and then to run the system as usual. The answers are usually not translated, which is, especially for factoid questions, often also not necessary. Furthermore, users may well be able to browse through answers in another language (or may use on-line translation services to get a general understanding) but still feel more comfortable in asking in their own language. In our study, we will also follow this approach and, therefore, concentrate on the translation of questions. In particular, we take the case of English-to-Dutch question answering, mainly because of our interest in Dutch QA.

The simplest approach to cross-lingual QA is to use available on-line MT engines for the translation of questions (see for instance Larosa et al. (2005)). There are several problems with this approach: First of all, most of these engines are general-purpose MT systems which are not optimized for the specific task of translating questions. However, questions have a very specific syntactic structure, often very different from other sentence types. They often show similar patterns, especially factoid questions, which makes them suitable for a data-driven approach modeling these specific patterns in particular.

Another problem with on-line services is their availability and reliability. A cross-lingual QA system using such services always depends on these external resources and has to adjust to service changes and quality differences. For example, we experimented with Google Translate and observed differences in its behavior from one day to another, which seriously affected our QA pro-

Proceedings of the 13th Annual Conference of the EAMT, pages 112–119,
Barcelona, May 2009

totype: Google started to treat names (unknown to the system) in an unpredictable way, for instance, translating the German city name "Wernigerode" into "Waterloo". This, of course, has severe consequences for a QA system trying to answer questions such as "Where is Wernigerode?" or "How many people live in Wernigerode?". Later, however, this behavior was corrected by Google.

To sum-up: in order to build a simple cross-lingual QA system one needs a proper translation component. In order to reduce the dependency on on-line services and, especially, in order to improve translation quality we like to develop a task-specific translation component for our system. In this study we investigate if we can use standard techniques for doing this. Especially we like to see how far we can get with extremely scarce resources when optimizing with linguistic features and additional resources such as term databases.

The paper is organized as follows: In the next section we present the general setup including a brief discussion of the baseline and the approaches applied. Thereafter, various experiments are presented, and, finally we summarize our findings with some conclusions.

2 General Setup

In our experiments, we focus on factoid questions, especially the ones used at the QA tracks at CLEF. This is mainly due to the availability of data for training and testing. In the following some more details of the data collected are given. Thereafter, we briefly summarize the baseline scores using Google Translate and and the MT approaches applied in the experiments below.

2.1 Data

Cross-lingual QA has been a shared task at CLEF for several years. There are various multilingual resources available via CLEF which we are grateful for. In particular, we use the Multi-eight-04 corpus, a collection of 700 questions in eight languages (Magnini et al., 2005), the DISEQuA corpus, a collection of Dutch, Italian, Spanish and English questions (Magnini et al., 2003a), and the Multi-six corpus, a collection of 200 questions in six languages collected from CLEF QA-2003 (Magnini et al., 2003b). Altogether, this amounts to 1349 questions with English and Dutch translations.

Additionally, we also have one source of Dutch questions coming from the popular Winkler-Prins game (a Dutch quiz game similar to Trivial Pursuit), which we have used previously for training our disambiguation module when parsing questions (Bouma et al., 2005). This monolingual corpus contains 4509 questions.

Another resource that we use is the multilingual Europarl corpus (Koehn, 2005) with its more than 1,000,000 parallel sentences. From this corpus we also extracted 31,506 questions by simply searching for lines ending with question marks in both, English and Dutch translations. Certainly, these are not the typical questions to be expected as input for a QA system. However, they are still useful as they represent the specific syntactic structures of questions.

Finally, we also collected multilingual term databases from Wikipedia and Geonames.org. From the latter, we simply extracted all pairs of Dutch and English place names giving us 55,381 entries. From Wikipedia we made use of the link structure between Dutch and English pages and extracted 145,510 pairs of Wikipedia lemmas.

2.2 Baselines

The baseline refers to the approach of applying available general purpose MT engines. We have chosen to use the popular service by Google (Goo, 2008). For evaluation purposes we took 100 questions from our parallel data which will be applied in all experiments below. We are aware that this test set is very small but we had to compromise due to the size of the material available to us. Table 1 shows the BLEU scores[1] obtained when translating our test set with Google (English to Dutch) and scoring with the one reference translation per question given in the data.

Google Translate	BLEU
October 2008	31.09
November 2008	32.66
January 2009	32.45

Table 1: Translating the test set of 100 English questions to Dutch (on three different dates).

The "Google" baseline can be seen as a very strong baseline as it is a running system with many satisfied users. Also, manually inspecting the translations show that the quality is reasonable

[1]All BLEU scores are computed using the `multi-bleu.perl` script from the Moses package.

and most of the translated questions are indeed correct or at least understandable.

In table 1 above we can see the general problem of on-line services as we have discussed earlier. The system is in development and its behavior is not stable. Although, the BLEU scores are not very different, the output can vary quite a lot. In the introduction we already mentioned the issue of wrongly translating place names at some point. In the version of October 2008 we observed another issue which is quite important for our QA system: Google Translate did not recognize several Wh-words correctly but translated them as relative pronouns. Consider the following examples:

English: Who is the Prime Minister of Ireland ?
Dutch: *Die* is de premier van Ierland ?

English: When was Elvis Presley' s first record recorded ?
Dutch: *Toen* was Elvis Presley's eerste record geregistreerd?

English: For which film did Robert Bresson win the Grand Prix at Cannes ?
Dutch: Voor *die* film deed Robert Bresson wint de Grand Prix in Cannes ?

This, of course, is a serious problem for a QA system that uses patterns involving Wh-words in its question analysis when looking for the question focus. However, this problem seems to be solved in later versions of the on-line engine.

2.3 PSMT for Question Translation

Phrase-based statistical machine translation (PSMT) is currently extremely popular and can be seen as one of the state-of-the-art approaches in today's machine translation research. Its popularity is also due to the availability of tools for building statistical models (word aligners and phrase extractors) and for the actual translation (decoders). The techniques are becoming so well-known that we omit the general introduction of the (P)SMT approach and just refer to standard literature (see for example (Brown et al., 1993; Och and Ney, 2003; Koehn et al., 2007; Koehn and Hoang, 2007)).

It might come as a surprise to see PSMT as one of the approaches applied here after the introduction of our training data. SMT usually requires large amounts of training data (for instance, more than 1,000,000 sentences of parallel data). However, for our task-specific approach we only have a tiny amount of translated questions available. On the other hand, we know that questions (especially factoid questions) follow very regular patterns. They are often very short and usually do not include embedded clauses or other complex structures. The general question we want to ask here is: Can a small amount of very regular, task-specific training data be used for training a statistical model that can compete with larger models? We also like to know how far we can get when adding additional resources and tweaking the system in such a way that it maximizes the performance possible with the data available. Finally, we also want to see the effect of domain/task-specific data when combined with out-of-domain data, also in comparison with our strong baseline.

In our experiments we apply the Moses system (Koehn et al., 2007) and its accompanying tools such as GIZA++ (Och and Ney, 2003) and IRSTLM (Frederico et al., 2008). We mainly use standard settings for all components if not stated otherwise.

2.4 EBMT using Proportional Analogies

The idea of example-base machine translation (EBMT) using proportional analogies has been introduced by Lepage and Denoual(2005). The idea is to solve string-level analogies in order to translate new sentences given a database of example translations. For this no pre-processing, sentence decomposition, word level alignment nor any other type of training or generalization is needed. The translation process entirely relies on solving analogical equations. Proportional analogies are denoted as $A : B :: C : D$, which is to be read as "A is to B as C is to D". An example of such an analogy is given in figure 1.

It walk s	:	It walk ed	::	It float s	:	It float ed
across the		across the		across the		across the
str e et .		str e et .		riv e r .		riv e r .

Figure 1: An example of a proportional analogy.

In a parallel corpus all sentences are aligned to corresponding translations in another language. Proportional analogies in the source language can now be used to identify existing entries in the corpus for sentences that are not part of the corpus. Corresponding analogical equations on the target language side can then be used to actually find the translation of these new incoming sentences. The following summarizes the translation process:

example database: $X = (X_{src}||X_{trg})$
input sentence: D (to be translated)

1. $\forall A_i, B_i \in X_{src}$ solve $A_i : B_i :: x : D$

2. $\forall x = C_{i,j}$ solve $\widehat{A_{ik}} : \widehat{B_{ik}} :: \widehat{C_{i,j}}^{k} : y$ where $(\left[S_i, \widehat{S_{i,k}}\right] \in X)$[2]

3. sort solutions $y = \widehat{D_{i,j}}^{k,l}$ by frequency[3]

recursion: possible after step 1: $\forall x = C_{i,j} \notin X_{src}$ translate them with the procedure above and add solutions to the corpus X

This approach has been successfully applied to various language pairs in a specific domain (travel and tourism). For more details we refer to the background literature (Lepage and Denoual, 2005).

There are several reasons why this approach is quite appealing for our task, the translation of questions. Firstly, it does not use a statistical model and, therefore, does not require huge amounts of training data with similar contents to get reliable counts. (However we still require a good amount of examples to obtain a reasonable coverage.) Secondly, we believe that this approach works best with rather short sentences for which reasonable analogies can be found in a limited amount of time. The questions we deal with are rather short and, therefore, seem to be appropriate. Thirdly, the parallel database can easily be extended by other translation data, for example, databases of translated terms. As we know, factoid questions often use very regular structural patterns. Simple analogies can be used to replace, for example, named entities that are part of the question focus. Consider the following, very simple example:

- input sentence: 'What is the capital of Armenia?'
- bitext:

What is the capital of Somalia?
Wat is de hoofdstad van Somalië?
Flag of Armenia
Vlag van Armenië
Flag of Somalia
Vlag van Somalië

Flag of Somalia	: What is the capital of Somalia?	::	**Flag of Armenia**	: What is the capital of Armenia?
Vlag van Somalië	: What is de hoofdstad van Somalië?	::	Vlag van Armenië	: **Wat is de hoofdstad van Armenië?**

For our experiments we will use our small set of example questions augmented with the term databases extracted from Geonames and from Wikipedia. However, considering the size of our example corpus of questions we do not expect to find many solutions using this approach but the ones found are expected to be highly accurate.

[2]Indeces j, k, l are added to indicate that there are several solutions possible when solving analogical equations.

[3]The same solution can be often be found in various ways with the procedure above. Frequency is assumed to be a good indicator for preference.

3 Experiments

We will now turn to the actual experiments. We will use the data as described in the previous section, in particular, we will use the same evaluation set of 100 questions for all experiments listed below. Furthermore, for training the PSMT models we will use a development set of 100 questions taken from the training data for tuning model parameters with minimum error rate training. In the following we will look at individual experiments. A summary of our results is shown in section 4.

3.1 Different Types of Training Data

In the first experiment we compare PSMT models trained on our tiny task-specific corpus with one trained on much larger material (namely data from the Europarl corpus). Table 2 shows the BLEU scores obtained after training and tuning with standard settings of the Moses system.

language model & translation model	BLEU
CLEF	26.60
CLEF+terms	28.53
EP	27.20

Table 2: BLEU scores for PSMT models trained on tiny task-specific data (CLEF) and on larger parallel training data (Europarl = *EP*); *terms* refers to Wikipedia lemmas and Geonames

In all settings we use the target language part of the parallel training data for estimating the language model probabilities. We can see that the tiny model almost performs as well as the one trained on much larger material according to BLEU scores. The tiny model suffers a lot from unknown words, among them many named entities. Adding Wikipedia lemmas and translated Geonames improves the system a lot and the performance even passes the larger model now.

In a second experiment we like to investigate the influence of the language model. In particular, we like to see how important is the use of appropriate data when building the language model. Table 3 summarizes our results.

	translation model		
language model	CLEF	+terms	EP
EPq	27.94	27.73	28.33
CLEF+EPq	28.59	30.30	30.49
CLEF+WP+EPq	28.91	30.36	31.10
CLEF+terms+WP+EPq	29.51	31.86	30.08

Table 3: Different language models for basic PSMT settings. *EPq* refers to questions from the Europarl corpus.

As we can see, the performance improves significantly when using language models consisting of questions only (except the terms added in the last setting). We argued already earlier that a system should make use of the specific syntactic patterns of questions and the results in table 3 demonstrate the success of adapting the language model (which is mainly responsible for grammaticality and fluency in the target language) to this kind of data. Observe that the language model using Europarl questions outperforms the one estimated on the entire Europarl corpus when combined with the translation model from the same corpus. Adding small amounts of task-specific data (CLEF) improves the scores even further.

Finally, we like to see the influence of task-specific training data for estimating both, translation model and language model, when combined with larger out-of-domain data. Table 4 shows the BLEU scores obtained for various data sets.

	translation model	
language model	CLEF+EP	CLEF+terms+EP
CLEF+EP	33.27	33.76
CLEF+EPq	36.34	35.21
CLEF+WP+EPq	36.79	34.76

Table 4: Combining task-specific and out-of-domain data.

The results show clearly that it is still helpful to add more data when building statistical MT models. However, in-domain data (even tiny amounts) are very important also for the translation model as we can see in the BLEU scores above. The results are all above the previous ones and now also exceed the Google baselines. Note that the term databases do not add anything to the model anymore when combining the CLEF data with the larger Europarl corpus. This probably means that the necessary terms are already included in the data and further databases are not necessary.

Finally, we want to mention that we also tried to use various combinations of in-domain and out-of-domain models (language models and phrase tables) as separate factors in the log-linear PSMT model and alternative paths during decoding. However, after minimum error rate tuning the scores were similar or below the ones presented above and, therefore, we omit these experiments.

3.2 Factored Models

One of the important extensions in the Moses system is the support of so-called factored translation models (Koehn and Hoang, 2007). It actually provides a framework for the integration of linguistic features or any other word-level features to be integrated in the translation models, the language models to be used in combination by the Moses decoder. There are many possibilities for an integration of such extra features. For example, a phrase table can combine surface word forms with POS tags and translate them into corresponding word roots with attached POS labels for the target language. Factors can also be translated separately using different phrase tables. They can even be generated on the target language side from other factors. In this way translation decisions can be based on various factors allowing different kinds of generalizations, sparseness of data can be reduced for example by the use of lemmas instead of word forms together with a target language generation step and fluency of the output can be improved by the integration of language models over different features.

In order to test various settings using factored models we parsed our data with Alpino (van Noord, 2006) on the Dutch side and extracted word-level features from the dependency graphs created by the parser. In this way we got the following factors: root forms, coarse POS tags, fine-grained POS tags with morphosyntactic information and dependency relations (to the corresponding head word). For English we used the C&C tools (Curran et al., 2007) to tag the data directly with POS tags and CCG supertags. After doing this we ran various experiments with different settings for factors and translation and generation steps. Unfortunately, the results so far are quite disappointing. We omit most of our results and just list a few of the better example in table 5 below.

Unfortunately, no significant and consistent im-

	translation model	
LM = CLEF+EPq	CLEF	CLEF+EP
baseline	28.59	36.34
w→ l,p+c→ p+r,l+p+r→ w	27.97	33.31
w→ l,p→ p,l+p→ w	29.16	30.14
w→ w, generate p	28.90	36.18

Table 5: Example settings of factored PSMT trained on CLEF and CLEF+EP (w=wordform, l=lemma, r=dependency relation to head, p=POS)

provements could be measured with the settings shown in the table. The first setting refers to a model with two translation steps (words to lemmas and POS-tags+CCG supertags to POS tags and dependency relations) and one generation step (lemmas+POS tags+dependency relations to surface word forms). The second setting refers to a model with also two translation steps (word to lemmas and POS tags to POS tags) and a generation step (lemma+POS to surface words). Unfortunately, the performance drops for all settings.

The last setting in table 5 refer to a standard approach (word to word translation) with a generation step added to generate POS tags. The reason for doing this is to add a POS language model into the decoding process for better generalization. However, no consistent improvement can be seen here.

Similar behavior could be observed for other kinds of factored models we have tried so far. In most cases we observed decreasing performances. More investigations are required to get a clear picture of the capability of factored translation models.

3.3 Escaping Named Entities

We already mentioned earlier that questions follow similar patterns and often differ only in certain named entities being part of the question focus. One idea is to escape the named entities from the statistical model and to translate them separately in a second step. Here again, we are interested in how far we can get with small amounts of training data. Replacing named entities with a dummy variable modifies the training material in such a way that these regular patterns should be more visible for a statistical approach and, thus, the model should become more general.

We used the following procedure: First we replaced named entities (NE) with a special dummy word and trained the PSMT models on the modified data. Here we used a very simplistic approach to detect NE's by replacing all (sequences of) capitalized words with the dummy word in source and target language. In the translation step we simply applied the models as usual and thereafter replaced dummy words in the output with name translations from our Wikipedia/Geonames database. Unknown names are simply copied as usual and if there are less variables in the output than in the input we added the names at the end of the translated question. This is certainly a very simplified procedure and only of conceptual interest. The results of applying this approach are shown in table 6.

	translation model	
language model	CLEF	CLEF+EP
CLEF	30.40 (32.80)	28.37 (29.85)
CLEF+EPq	30.74 (34.33)	35.21 (39.43)
CLEF+WP+EPq	32.41 (35.77)	35.29 (39.73)

Table 6: PSMT models with escaped named entities. Scores in brackets are BLEU scores without considering the actual NE translation.

As we can see, we can further improve the models using only our tiny amount of parallel training data and obtain scores comparable to the Google baselines. On the other hand, for the combined training data we can see a negative effect of this approach. However, as the scores in brackets show, improvement might be possible with a more sophisticated NE detection and translation procedure. These scores are measured on translating the "NE templates" only without replacing variables with corresponding names and, therefore, can be seen as upper bounds for this method.

3.4 Source Language Reordering

Yet another idea for improving our models is to apply source language reordering techniques before training and translating. This is especially important in our case when translating English questions where the predicate is often split into an auxiliary and the infinite main verb is moved to the end of the question. That this is a serious problem could be seen at the following translations obtained by the models from the previous section:

When did Armenia become independent ?
* Wanneer stierf Armenia onafhankelijk ?
(* *When died Armenia independent ?*)

This error appears of course not because "did" and "die" are so similar to each other but because there are apparently many questions about people's deaths in our training data. The preference for links at similar positions causes the word aligner to select "did" as the alignment of "stierf", which in the end causes the error described above.

When did Shapour Bakhtiar die ?
Wanneer stierf Shapour Bakhtiar ?
At what age did Fernando Rey die ?
Op welke leeftijd stierf Fernando Rey ?

The success of "pre-ordering" the source language has already been shown in earlier studies (Collins et al., 2005) and also moving the main verb in questions has been applied in other studies (Nießen and Ney, 2004). We therefore parsed our data with the Stanford parser obtaining not only phrase-structure trees but also dependency relations. We then moved the infinitive next to the auxiliary if they are in a (corresponding) direct relation to each other:

original: How did Jimi Hendrix die ?
reordered: How did die Jimi Hendrix ?

original: What language do the Berbers speak ?
reordered: What language do speak the Berbers ?

This is done for the CLEF questions before estimating the MT models and before translating questions from the test set. Results using this approach are shown in table 7.

	translation model	
language model	CLEF	CLEF+EP
CLEF	29.39	26.93
CLEF+EPq	33.18	38.07
CLEF+WP+EPq	33.58	37.46

Table 7: Simple re-ordering of the source language questions.

As we can see, the BLEU scores improve significantly for both, the small and the combined training data. Even for our small training set we now obtain scores above the Google baseline and for the combined data set we are more than five points ahead. This is very encouraging and further investigations in this direction should be carried out in future.

3.5 Analogical EBMT

Finally, we also want to look at the alternative approach of analogical learning for example-based MT. The approach has been briefly discussed earlier. We now apply it using the software of Lepage[4] and the CLEF questions together with our bilingual term database as example corpus. As expected the coverage of our examples is not sufficient. Only a small fraction could be translated using this technique (10 questions out of 100). It is not worth mentioning the BLEU score for the entire test set (the EBMT system functions as a translation memory returning the closest match in cases of failures of the analogical procedure). However, for the actual translations the accuracy in terms of BLEU scores is very high (70.7 BLEU). For a comparison, the best model so far from the previous sections scores only 66.1 BLEU on the same questions. It is therefore worthwhile considering this approach especially if the training corpus could be extended in future. We also experimented with a simple backoff approach in which we use the two-step procedure from section 3.3 together with the analogical EBMT approach in cases where the analogical solver did not succeed to find a translation of the original question. Using this strategy the number of translated questions goes up to 24. However, the BLEU score drops significantly to about 53.

4 Discussion & Conclusions

In this paper we addressed the task of translating questions for cross-lingual question answering. The motivation of this study is the development of a task-specific component that outperforms a general purpose engine. For this we used standard approaches to statistical MT with a mixture of task-specific data and out-of-domain data. One important aspect of our experiments is to test possibilities of building data-driven translation models from extremely scarce resources. Several techniques have been used ranging from source language reordering to named entity escaping and factored models with linguistic features. A summary of our results is shown in table 8.

According to the automatic measures on a small test set we succeeded to outperform a strong baseline given by a state-of-the-art general purpose translation engine (Google Translate). However, human evaluation should be performed in future to support the automatic evaluation. Furthermore, another look at the integration of linguistic features is also on our research agenda. Finally, we would

[4]We are very grateful for making this software available to us.

CLEF: 26.60, Europarl: 27.20, Google: 32.45

CLEF + extensions	
+ terms	28.53
+ terms & Q-LM	31.86
+ terms & Q-LM & escape NE	32.41
+ Q-LM & source-reordering	**33.58**
CLEF + EP + extensions	
Q-LM	**36.79**
Q-LM & factored	**36.18**
Q-LM & escape NE	**35.29**
Q-LM & source-reordering	**38.07**

Analogical EBMT (for 10 out of 100) = 70.67
backoff EBMT/NE (for 24 out of 100) = 53.06

Table 8: Summary of experiments. *Q-LM* refers to the language model trained on questions. Scores in bold denote results above the Google baseline.

also like to combine the strengths of the various approaches in order to build a system with better performance. Initial experiments with simple backoff strategies have already shown encouraging results.

References

Bouma, Gosse, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jörg Tiedemann. 2005. Linguistic knowledge and question answering. *Traitement Automatique des Langues (TAL)*, 3.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistcal Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June.

Collins, Michael, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540, Morristown, NJ, USA. Association for Computational Linguistics.

Curran, James R., Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the ACL 2007 Demonstrations Session (ACL-07 demo)*, pages 29–32, Prague, Czech Republic.

Frederico, M., N. Bertoldi, and M. Cettelo, 2008. *IRSTLM Language Modeling Toolkit, Version 5.10.00*. FBK-irst, Trento, Italy.

2008. Google translate. http://translate.google.com/.

Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Prague, Czech Republic, June.

Koehn, Philipp, 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. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the ACL, demonstration session*, Prague, Czech Republic.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Larosa, S., J. Pearrubia, P. Rosso P., and M. Montes. 2005. Cross-language question answering: The key role of translation. In *Proc. Avances en la Ciencia de la Computacin, VI ENCuentro Int. de Computacin, ENC-2005*, pages 131–135, Puebla, Mexico.

Lepage, Yves and Etienne Denoual. 2005. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, 19(3-4):251–282.

Magnini, B., S. Romagnoli, A. Vallin, J. Herrera, A. Peas, V. Peinado, F. Verdejo, and M. de Rijke. 2003a. Creating the disequa corpus: a test set for multilingual question answering. In Peters, Carol, editor, *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway.

Magnini, B., S. Romagnoli, A. Vallin, J. Herrera, A. Peas, V. Peinado, F. Verdejo, and M. de Rijke. 2003b. The multiple language question answering track at CLEF 2003. In Peters, Carol, editor, *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway.

Magnini, B., A. Vallin, C. Ayache, G. Erbach A. Peas, M. de Rijke, P. Rocha, K. Simov, and R. Sutcliffe. 2005. Overview of the CLEF 2004 multilingual question answering track. In Peters, C., P. D. Clough, G. J. F. Jones, J. Gonzalo, M. Kluck, and B. Magnini, editors, *Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign*, volume 3491 of *Lecture Notes in Computer Science*. Springer.

Nießen, Sonja and Hermann Ney. 2004. Statistical machine translation with scarce resources using morpho-syntactic information. *Computational Linguistics*, 30(2):181–204, June.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

van Noord, Gertjan. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.