Literature survey on comparable corpora

by

Subhash Iyer
Roll No: 133050018

Under the guidance of

Prof. Pushpak Bhattacharya

Department of Computer Science & Engineering,
Indian Institute of Technology, Bombay

Contents

Chapter 1

Introduction

Machine translation(MT) is the task of automatically translating a text from one language into another. While translating a text from one language to another, the quality of translation can be broadly measured based on two factors- adequacy and fluency. Adequacy measures the extent to which the meaning is transferred while translating from one language to another while fluency measures the correctness of formation of sentence after translating.

In this chapter, we discuss various approaches to machine translation that currently exists on a top level. Also, we discuss the theory of statistical machine translation and why they require large parallel corpora. Further, in this chapter we also classify different bilingual corpora based on their comparability and the standard concepts that are applicable in finding translational equivalents from them.

1.1 Approaches to machine translation

There are various approaches using which machine translation can be performed which are as described in following subsections.

1. **Rule Based MT**
   A set of rules are defined for translation from one language to other which are used to translate the text. The rules may be defined using the linguistic information about the source and target languages covering semantic, syntactic and morphological features of languages.

2. **Direct MT**
   This machine translation system uses a dictionary lookup and substitutes words from one language to another language. This approach can be found useful for languages which follow same word order. But,

languages that have different word order will perform very poorly in this system.

3. **Transfer Based MT**
   Transfer based approach converts a text from one language into an intermediate representation that captures the structure of the text and then uses that representation to translate into target language. This uses lexical, syntactical and morphological features to extract an intermediate structure and then transfer into other language.

4. **Interlingua based MT**
   Interlingua based MT is similar to transfer based MT where a intermediate representation is created. But, in transfer based MT, the intermediate representation is dependent on the two languages in question whereas interlingua based MT aims to convert into a representation which is independent of the two languages.

5. **Statistical MT**
   Statistical MT uses statistical methods to translate a text from one language to another. It uses a parallel corpus which contains a collection of sentences in one language and sentences in other language which are exact translations of the other.

6. **Example based MT**
   This approach uses previously translated sentences for purpose of translation. This method divides a text into smaller texts and look up how these smaller texts were translated in previously translated sentences and use this to translate the complete sentence.

7. **Hybrid MT**
   Hybrid MT makes use of combination of above approaches to perform translation. They make use of both rule and statistics while translating from one language to other.

1.2 Statistical MT

In statistical MT, translations are generated using statistical models whose parameters are derived from a parallel corpus. The basis of statistical MT is that we can model the task of machine translation statistically as $P(e|f)$ where e is the sentence in target language while f is the source language sentence. This is the measure of probability that a source sentence f is translated into target sentence e. There are various methods using which the

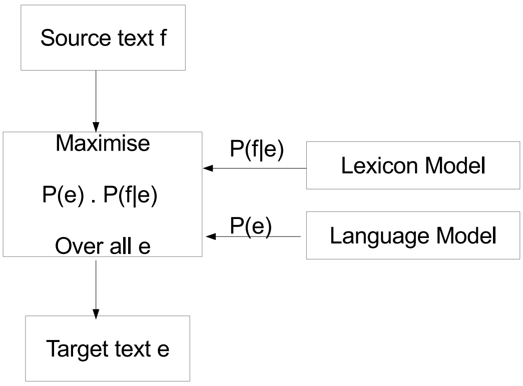

Figure 1.1: Noisy channel model of SMT

translation is modeled. One of the formulation is using the noisy channel model, which is described below:

$$P(e|f) = \frac{P(e)P(f|e)}{P(f)}$$

Thus, the translation e for a given sentence f is given by,

$$e^* = \underset{e}{argmax} P(e|f) = \underset{e}{argmax} P(e)P(f|e)$$

Here P(f) in the denominator is ignored under argmax computation since it is constant for all e. Also, note that now P(e) models the fluency of the translation which is obtained by training a language model while $P(f|e)$ models the adequacy of the translation which is obtained by training lexicon model. Figure 1.1 shows the noisy channel modeling of statistical machine translation.

1.3 Importance of large parallel corpora for SMT

Statistical machine translation uses parallel corpora heavily for translation. It uses corpora to estimate various statistical parameters like translation probabilities. The quality of translation heavily depends on the quality of parameters learnt from the corpora. Since, the size of vocabulary of language and set of sentences is very large, it requires very large data to reliably estimate the parameters. Thus, larger the size of parallel corpus, better will be the quality of translation. But, parallel corpora are not readily available for most language pairs. It requires manual labour to create parallel corpora which is a very labour intensive work. So, we need methods by which we can create parallel corpora automatically with minimal efforts. For such a task, comparable corpora comes in handy. Comparable corpora contains documents that convey same information in different languages. Thus, by exploiting comparable corpora, one can aim to extract some parallel sentences from them. This will increase the size of available parallel corpora and thus produce good quality translations.

1.4 Types of bilingual corpora

Any bilingual corpora can be classified broadly into four types based on the type of information contained in the two corpora (Fung and Cheung, 2004)

1. Parallel corpora
   This is a sentence aligned corpora containing bilingual translations of the same document. This is the type of corpora that one needs for statistical machine translation. Such type of corpora need to be created manually and hence, they are not readily available. Examples of such corpora are Hong Kong Law corpus which contains English-Chinese parallel sentences.

2. Noisy parallel
   This contains non-aligned sentences that are nevertheless mostly bilingual translations of the same document. Such documents are on the same topic and hence, contains sentences that are roughly the translations of each other with some deletions and insertions of some sections of documents. An example of such a corpus will be the news articles published by various news agencies.

3. Comparable
   This contains non-sentence-aligned, non-translated bilingual documents that are topic-aligned. An example of such a corpus can be again news articles that are published within a time frame.

4. Very non-parallel corpora/ quasi comparable corpora
   This contains far more disparate, very-non-parallel bilingual documents that could either be on the same topic or not. An example of such a corpus will be TDT3 corpus which contains transcriptions of various news from radio broadcasting or TV news report. Such a kind of corpus

contains some parallel sentences, but most of the sentences turn out to be paraphrases or are non-translations of each other.

1.5 Common rules for parallel sentence extraction

There are some very common rules that are applied while extracting parallel sentences which are described below (Fung and Cheung, 2004). Most of the systems of parallel text mining try to incorporate some or all of these features while building their system.

1. Length of sentences:
   It is observed that generally, the length of a sentence in one language and its translation in other language tend to be similar. Thus, by looking at the lengths of the two sentences, one can make a guess if the sentence pair might be parallel to each other.

2. Position of sentences in documents
   Sentences are assumed to correspond to those roughly at the same position in the other language. This is based on the idea that when two documents are describing the same topic, the sequence in which the two documents describe sub-topics will be similar and hence, their positions should be similar. But, this assumption does not hold true for quasi-comparable corpora as they are not topic aligned.

3. Word overlap
   Word overlap is defined as the number of words in a source sentence that have a corresponding translation in other language. A pair of bilingual sentences which contain more words that are translations of each other tend to be translations themselves.

4. Frequency of word pairs
   Occurrence frequencies of bilingual word pairs in the two languages are similar. This again stems from the fact that the two documents are topic aligned and hence, the assumption does not hold for quasi comparable corpora.

5. Word sense
   Words have one dominant sense per corpus. This is again true for only topic aligned document pairs. Since, the two documents are on same topic, the words will be used in a sense specific to that topic. This

assumption in turn implies that words have a single translation per corpus.

6. Context of words
   Sentences generally have words that occur with same set of words in different language pairs. This forms the basis of calculation of similarity among various document pairs during extraction of parallel sentences.

7. Document similarity
   Parallel sentences are more likely to exist in document pairs with high similarity scores. This forms the basis for filtering out non-comparable document pairs before extraction of parallel sentences.

Out of the rules described above, only word overlap measure and length of sentences are the ones that can be applied across wide range of corpus. Applicability of other principles like context, position, or word frequency depends on the extent of comparability of the corpus.

1.6 Outline

In this report we present various existing approaches for extraction of parallel corpora from comparable corpora at different levels of granularity like sentence level, phrase level and word level.

1.7 Summary

This chapter gives an overview of different approaches to machine translation. This is followed by description on statistical machine translation and need for large parallel corpora for same. It introduces different types of bilingual corpora that exists and how they differ in their comparability. It also gives a brief idea about the common principles that are employed while extracting parallel text from a comparable corpora.

Chapter 2

Parallel sentence extraction

Traditionally, the focus of research in mining from comparable corpora has been on sentences and lexicons. The basic premise of such a focus was that comparable documents are likely to contain parallel sentences. But, in reality this assumption is only valid for noisy parallel and comparable corpora described in previous chapters. Parallel sentences are less likely to exist in a quasi-comparable corpora. Following we describe some state of the art work for extracting parallel sentences.

2.1 Maximum entropy classifier

The basic intuition behind this approach is that the task of finding parallel sentences from a comparable corpus is equivalent to the task of classifying a pair of sentences as parallel or non-parallel. Thus, we can train a binary classifier for the same that, given a sentence pair, can identify whether a sentence is a translation of other.

(Munteanu and Marcu, 2005) use a maximum entropy classifier for the task of extracting parallel sentences. For a given pair of sentences, a set of features are extracted and these features are chosen in such a way that they give a clear indication of whether the two sentences are parallel. The model defined is a log-linear combination of the feature functions which tries to maximize the entropy of the system. Thus, we have the following expression used for classification,

$$P(c|sp) = \frac{1}{Z(sp)} \prod_{j=1}^{k} \lambda_j^{f_j(c,sp)}$$

where c is the class (parallel or not parallel), $Z(sp)$ is normalization factor, λ_j are parameter weights and f_j are the feature functions defined on the sentence pair *sp*. The classifier assigns a weights λ_j to each feature j used in for classification. During training, these feature weights are learned such that they maximize the entropy of the system. The sentence pair that have classification score above a threshold can be considered parallel.

For the task of classification, a set of features are defined based on the word-level alignments between the sentence pair in addition to some general features. The features used can be listed as follows:

General features

1. Length of sentences:
   There is general observation that sentences that are translations of each other are roughly of same length. Thus, features based on length difference, length ratio are defined

2. Percentage of words that have translation on other side:
   Using a bilingual dictionary, a set of words are identified that are translation of each other. Larger the number of words that have translation on other side, more likely the sentences are translations of each other.

Alignment features

A word level alignment is trained on a given seed parallel corpus and this is used to find most likely alignment for a given sentence pair. Following this, a set of features are defined based on the alignment.

1. Percentage and number of aligned words:
   This is based on the idea that parallel sentences tend to have high number of words that are aligned by the alignment algorithm used, as compared to non-parallel sentence. Thus, higher number of aligned words, more likely is that the sentences are parallel.

2. Fertility:
   Fertility of a word in an alignment is defined as the number of words it is connected to. In general, one can expect a word to be aligned to a small number of words (less than 3) on the other side if the two sentences are parallel. Thus, if a word is being aligned to large number of words, then it indicates non-parallelism of the sentence pair. In general, in a parallel sentence pair, fertility of words is usually low and also, number of words with high fertility is low. Hence, the top three largest fertilities of words are used as feature for classification.

3. Longest contiguous span:
   A contiguous span is a sequence of words in one language that are translated to a sequence of words in other language. A span may

contain words with no translation on other side, but cannot contain words that have translation outside the span. Parallel sentences usually tend to contain contiguous words in one language that are translated into a contiguous set of words in other language. Thus, if a sentence pair contains large contiguous span of words, then they are translations of each other

4. Alignment score:
   Alignment score is computed as product of translation probabilities of each aligned word in the sentence pair. Aligned words in non-parallel sentence pairs will have low probabilities. Thus, a high value of alignment score is indicative of parallelism.

2.2 Iterative mining

One of the basic resources required for the extraction of parallel sentences is the presence of a bilingual dictionary. Bilingual dictionary plays a major role in finding out documents which are likely to contain parallel sentences as this step involves glossing words in one language into another for finding their similarity. Hence, one can safely say that document pair extraction is heavily dependent on the coverage of the dictionary. In the absence of any word translation in the dictionary, one might end up marking a document as not parallel even though they might be actually parallel. By use of iterative mining, the goal is to update the bilingual dictionary as and when we obtain any new word translations from the extracted parallel sentences and reiterate the process.

The technique is centered around the idea that better document matching leads to better parallel sentence extraction, better sentence matching also leads to better document matching (Fung and Cheung, 2004). This idea gives an iterative nature to the proposed approach. Another key idea in the algorithm is the “find one get more” principle (Fung and Cheung, 2004). This means that if a document pair contains a parallel sentence pair, then it is likely that it may contain more parallel sentences irrespective of whether the document pair has a good similarity score or not. This can be explained by the fact that sometimes the two documents describe about some common topic in a small section and then each document go on to describe different topics. Hence, if such a document pair is found, that pair is added to the selected document pair list for further processing.

The algorithm involves the following steps:

1. Initial document matching

For all documents in the collection, the target language is glossed using a bilingual dictionary. Following which, the two documents are represented as feature vectors and a similarity is computed between the two documents. If the similarity measure is below a threshold, the document pair is discarded. Similarity can be computed as a cosine similarity between document with feature weights as combinations of term frequency and inverse document frequency.

2. Sentence matching
   For each extracted document pair, a word vector is constructed for every sentence in the document. Then, for each sentence pair in the two documents, a similarity score is computed and those above a threshold are output as parallel.

3. EM lexical learning
   For all sentence pairs extracted by above step, lexicon translation probabilities of all word pairs in the sentence pair are calculated. If any new translation is found and is above a threshold, it is added to the existing bilingual dictionary.

4. Document rematching
   From the set of sentence pairs extracted from step 2, look for other documents judged to be dissimilar by step 1 that contain one or more of these sentence pairs. Following which other documents which are similar to those documents are extracted. This is based on the "find one get more" principle described above. Following which, the algorithm reiterates steps 2, 3 and 4 until convergence.

5. Convergence
   Sentence alignment scores and word alignments cores are computed at each step. The parameter values eventually converge and the extracted sentence pairs also converge to a fixed size. The alignment score measures on average, how many known bilingual word pairs actually cooccur in the extracted parallel sentences. Hence, it will converge to a fixed value once no new lexical translations or parallel sentences are found.

2.3 Grammar based approach

(Wu and Fung, 2005) propose the use of Inversion Transduction Grammar(ITG) for the purpose of extracting parallel sentences from comparable corpora. Their premise is based on that fact that two sentences are likely to have similar syntactic structure on both sides and hence, they try to exploit the structure of the sentence pair in two languages. ITGs are synchronous grammars where we have grammar rules where each non-terminal produces two outputs one for each language. Additionally, ITGs also allow inversion of production rules in two languages to be able to model the phenomenon of word re-ordering in different languages well. For example, a grammar rule $A \to [BC]$ will produce B_1C_1 in one language and B_2C_2 in other language whereas a rule $A \to< BC >$ will produce strings B_1C_1 in language 1 and C_2B_2 in other language. The symbols [ ] and <> are used to represent whether reordering takes place or not. Then, once we have all the grammar rules necessary for describing the two languages, a probability is assigned to each grammar rule and we create a stochastic ITG. Finally, for extracting parallel sentences, the two sentences are parsed using the stochastic ITG grammar constructed earlier and the best parse tree is constructed using the parsing algorithm described in (Wu, 1997). If the score of the obtained parse tree is above a threshold, then the two sentences are considered parallel, otherwise they are discarded.

2.4 Extraction as information retrieval

The system proposed by (Abdul Rauf and Schwenk, 2009) views the problem of extracting parallel sentences as the searching for information in source language in the target language using a information retrieval system and then use post filtering to extract meaningful sentences. The framework is that source sentence is first translated into target language using an existing machine translation system, The translated sentence is then posed as query to information retrieval system in target language over the documents that are known to be comparable to it and a set of n best matching sentences are output by the system. These sentences are the candidate parallel sentences for the source sentence. Now, filtering is performed the ranked sentences and the translated source sentence based on metrics like WER, TER and TERp. In evaluating these metrics, the input query is considered as the hypotheses and is compared against the output sentence for the number of edits required to match the two sentences. WER metric ensures that both the reference sentence and candidate sentence share common set of words. However two correct translations may differ in the order in which the words appear, which WER is unable to handle considering the fact that it looks at word level. This shortcoming is then overcome by the use of TER. Additionally, synonymous words and stemmed words are also looked for as a part of TERp metric.

2.5 Duplicate detection

(Jakob et al., 2010) views the problem as a task of detecting near duplicate sentences across languages. The approach used uses a shingling like approach to detect duplicates. Basic premise is to translate a source document into target language and then apply a duplicate detection algorithm on the translated and target documents to extract parallel sentences.

Initially, all source documents are translated into target language using a baseline machine translation system. They then make use of two class of n-grams namely, scoring ngrams and matching ngrams, which are extracted from the translated documents. The matching ngrams are used for extracting candidate document pairs and scoring ngrams are used to score sentence pairs. The system then extracts a set of documents that contain some matching ngram and then such documents are considered candidate document pairs. The system then computes pairwise scores for each document pair in the list and extracts n-best document list for each input document based on the score.

Scoring document pairs

The system makes use of scoring ngrams that were extracted earlier for scoring the document pairs. Let $F_d = \{f_1, f_2, \dots f_n\}$ and $F_{d^{`}} = \{f_1^{`}, f_2^{`}, \dots f_n^{`}\}$ be the set of scoring ngrams of documents d and $d^{`}$ respectively. Then interpreting $F_d$ and $F_{d^{`}}$ as incidence vector in the vector space of ngrams and using the inverse document frequency of ngrams as value of vector, the score between the two documents is computed as cosine similarity between the two vectors. Based on this score, n-best documents is extracted for each document and those pairs with scores lower than a threshold are discarded.

Sentence level alignment

After the extraction of n-best document list for each document, the next step used is to align sentences within documents. First sentence pairs in the documents are filtered based on the length of the two sentences and a probabilistic dictionary. Then, a more detailed score is computed between the sentence pairs as outlined below.

Let S be the set of source words, T the set of target words and $S \times T$ the set of ordered pairs. Let the source sentence contain words $S_0 \subset S$ and the target sentence contain words $T_0 \subset T$. An alignment $A_0 \subset S_0 \times T_0$ is found and scored as

$$score(A_0) = \sum_{(s,t)\in A_0} ln\frac{p(s,t)}{p(s)p(t)}$$

where the joint probabilities p(s, t) and marginal probabilities p(s), p(t) are taken to be the respective empirical distributions in an existing word aligned corpus. An alignment A is computed which maximizes the above score and if the score is above a threshold, the sentence pair is output as parallel.

2.6 Summary

In this chapter, we described various approaches that have been used for extracting parallel sentences from comparable corpora. The approaches described make use of various analogies to some existing approaches in information retrieval, text entailment, duplicate detection to identify parallel sentences. They also try to make use of word similarity, syntactic similarity(by use of grammar) to identify parallel sentences. In the next chapter, we describe various approaches for extracting parallel phrases.

Chapter 3

Phrase extraction

In many cases, exact parallel sentences may not exist in comparable documents. But, at the sub-sentential level, one can still find fragments of text that are parallel to each other. Taking into account this fact, various research has been performed to extract these phrases. This section gives a glimpse of various approaches proposed for the same.

3.1 Log likelihood ratio

The proposed method makes use of log likelihood ratio as a measure for comparison of the two texts (Munteanu and Marcu, 2006). This technique uses an approach inspired by signal processing which detects segments of source sentence that are translated into target segments. The key idea used in this technique is to find consecutive words in a sentence that have word translation probabilities above a threshold and occur in consecutive positions at the target side too. The technique makes use of two different probabilistic lexicons learned automatically from a seed parallel corpus GIZA++ lexicon and Log Likelihood Ratio(LLR). The GIZA++ lexicon is extracted from a seed parallel corpus with focus being on higher recall than precision.

Computing LLR score

LLR statistic in general sense gives a measure of the likelihood that two samples are not independent. In the context of identifying parallel phrases, LLR is used to estimate the independence of word pairs that occur in the corpus. A pair of words are said to be independent if they have are not translations of each other. If a target word f and source word e are independent, then $p(e|f) = p(e|\neg f) = p(e)$. LLR then gives measure of the likelihood of this hypotheses. A low value of LLR indicates that the distributions of $p(e|f)$

and $p(e|\neg f)$ are similar, thus implying that e and f are independent. Similarly, a high LLR implies that the two words are not independent. Also, a high LLR may be from a positive correspondence ($P(e|f) > P(e|\neg f)$ or a negative correspondence ($P(e|f) < P(e|\neg f)$). A positive correspondence gives a measure of two words being translations of each other while negative correspondences give a measure of words not being translations of each other.

LLR(e,f) is computed for every word e and f that are linked by the GIZA++ alignment obtained previously. This is then used to compute $p^{+}(e|f)$, the probability that target word f translates to source word e and $p^{-}(e|f)$, the probability that target word f does not translate to source word e. These distributions are obtained by normalizing $LLR^{+}(e,f)$ and $LLR^{-}(e,f)$ respectively over all values of e. A LLR score between words e and f are labeled as $LLR^{+}(e,f)$ if $P(e|f) > P(e|\neg f)$ and is labeled as $LLR^{-}(e,f)$ otherwise. Then, the above process is repeated for other direction also by swapping the roles of source and target language and $p(f|e)$ is computed.

Extracting parallel fragments

The algorithm proceeds forward by treating target sentence as a numeric signal. Now, after this formulation, translated words correspond to positive signals whose values are obtained from p^{+} distribution described previously and non-translated words correspond to negative signals whose values are obtained from p^{-} distribution. Each target word is linked with a word in the source that is most likely to be its translation. If there is no word aligned to the target word, a word in the source that is least likely to be not its translation i.e one with high value of negative association, is assigned a negative of that value in the signal or a value of -1 if there is no corresponding source word for that target word in the p^{-} distribution. This can now be considered as the initial signal. Now, in order to extract parallel phrases from this signal, an averaging filter is applied over the signal. The averaging is performed over a window of words that are adjoining the word of consideration. Then, the fragments of words with positive values after filtering are retained. This is then repeated in the other direction also and the resulting fragments of words in source and target side are considered parallel phrase pairs.

3.2 Chunking based approach

(Rajdeep et al., 2013) use an approach based on chunking where a document-aligned comparable corpus is used, and then they try to extract parallel chunks of texts from all possible sentence pairs. The sentences are first broken into fragments and then, the fragments are tested to determine which of them are actually parallel.

Instead of segmenting the source sentence into N-grams, chunking is used to obtain linguistic phrases from the source sentences. By use of chunks, they make sure that words within a chunk remain within the chunk after being translated. Thus, they ensure that there is no observed reordering outside the chunk while individual chunks may be reordered. Since chunks are actually linguistic phrases, it is possible to merge the translations of two chunks thereby producing a larger chunk. If the sentence is exactly parallel, one can obtain the entire sentence as final chunk after chunking. Such a phenomenon will not be possible by making use of n-grams.

Chunking Source Sentences and Merging Chunks

A CRF-based chunking algorithm is used to chunk the source side sentences. These chunks are further merged into bigger chunks, because sometimes, even merged bigger chunks can have a translation on the target side. In such a case, we can get a bigger parallel chunk. So, merging is done in two ways:

- **Strict Merging:** Merge two consecutive chunks only if they together form a bigger chunk of length <= 'V' words. 'V' can be an empirically decided value.

- **Window Merging:** In this type of merging, not just two, but as many smaller chunks are merged together, as possible, unless the number of tokens in the merged chunk does not exceed 'V'. Then, an imaginary window is slided over to the next chunk and the process is repeated.

Finding Parallel Chunks

To find parallel chunks, the source side chunks from the previous step are first translated to the target language using the baseline SMT system. Then, each of these translated chunks is compared with all the target side chunks of that document pair. The overlap between two target side chunks (one translated from source side chunk and the other is a chunk from the target side document) is found out. Here, the notion of overlap is:

$$\mathit{Overlap}(T_1, T_2) = \mathit{Number\ of\ tokens\ in}\ T_1\ \mathit{which\ are\ aligned\ in}\ T_2$$

The overlap of chunk is found both ways symmetrically, i.e., translated chunk to target side chunk and vice versa. If at least 70% overlap is found both ways, then the source side chunk corresponding to the translated chunk and the target side chunk are considered as parallel. Comparison of tokens for finding the overlap of two chunks is based on orthographic similarities like Levenshtein distance, longest common subsequence ratio and length of the two strings. Threshold for this matching is set empirically.

Refining the Extracted Parallel Chunks

From the extracted chunks, it is often observed that ordering of tokens in the source side is different to that of target side. Also, there could be some unaligned tokens on either side. So, the parallel chunk pairs are refined by reordering source side chunks according to its corresponding target side chunk and the unaligned tokens from either side are discarded.

3.3 Classifier based approach

(Gaizauskas, 2012) uses a SVM classifier based approach similar to the one used by (Munteanu and Marcu, 2005). But, here instead of extracting parallel sentences, the focus is to extract parallel phrases. Given a sentence pair, first all possible phrases of a given length are extracted and then consider all possible pairings between phrases as possible parallel phrases.

Training the classifier

In order to extract positive instances for training, it is essential that a seed parallel corpus be present. Then, a word alignment algorithm is run over such parallel corpus using a toolkit like GIZA++ to generate a phrase table. Then, all phrases contained in the phrase table are considered positive examples. Similarly, to get negative instances, for each sentence pair, all segments on the source side and target side with length within a range are identified. Then, each source segment is paired with each target fragment and finally, all those pairs which are not present in the phrase table are considered negative.

Features

They use following set of features in their experiments:

- Length difference in characters: It is the difference in number of characters in the source and target phrases.

- Length difference in words: It is is the difference in number of words in the source and target phrases.

- Number of words in phrase: It counts the number of words in source and target phrase each as separate features.

- First Word Translation Score: It indicates whether the first word in the source phrase is a translation of the first word in the target phrase. If this is the case, the translation probability is returned.

- Last word translation score: It indicates whether the last word in the source phrase is a translation of the last word in the target phrase. If this is the case, the translation probability is returned.

- Translation count: It is number of source phrase words which have translations in the target one and similarly in other direction.

- Translation ratio: It is ratio of the count of source phrase words which have translations in the target phrase and the number of words in the source language. This feature is measured in both directions.

- Is half translated: It is 1 if at least half of the source phrase words have translations in the target phrase, otherwise 0.

- Longest translated unit: It is count of words within the longest sequence of words which have all translations in the target phrase.

- Longest not translated unit: It is similar to the previous one but considers words which do not have translations.

- Translation position distance captures the distance between the source words positions and the position of their maximum likely translations in the target side. For each word in the source phrase we compute its translation position distance, sum all the distances together and return it.

3.4 Using hierarchical alignment model

(Jason and Daniel, 2012) use a existing alignment model, namely hierarchical model in this case, to extract parallel phrases. They first train a discriminative model which aligns words within a sentence. This model recursively scores individual alignments in a parse tree in a bottom up fashion. At the same time, for each node, it combines different alignments generated by their children thereby producing larger alignments. This process is continued until the parser covers the entire sentence. During the bottom up parsing, it also keeps track of back-pointers so that one can obtain the best derivation tree which maximizes the alignment score. This derivation tree gives a hierarchical partitioning of the alignment and the associated word-spans. Now, in order to extract parallel phrases, the tree is now parsed in a top down manner examining the fragments pointed to by in each node along with their scores. They then extract maximum length fragment with alignment score above a threshold as parallel fragment subject to following constraints.

1. The parent node in the derivation has a score less than threshold.

2. The length of the source span is greater than three.

3. There are no unaligned target words inside the fragment that are also aligned to source words outside the fragment.

Further, we observe that traversing the tree in a top down manner ensures that a larger fragment is detected first before its smaller variant. Also, the model stops traversing down the node, once a fragment has been identified from that node. This avoids any fragment which is a subset of extracted fragment from being extracted.

3.5 Generative model for sentence alignment

(Quirk et al., 2007) proposes two generative models to extract parallel fragments from comparable corpora. They have tried to model the phenomenon in comparable sentences that words may be inserted and deleted at any place in the two sentences depending on what information they include. One of the approaches proposed is by using a conditional model and other is by using a joint model. The approach is basically built upon the noisy channel modeling of the statistical machine translation system. In addition to the skeleton system of noisy channel model where a target sentence generates the source sentence, they augment an additional state where the current source word is generated from previous source word instead of from a target word and a probability is assigned for the transition from monolingual generation(generating source from source) to bilingual generation(generating source from target) and vice versa. Using this model, then the best alignment is computed using a viterbi like algorithm and phrases extracted from them. For extracting phrases from these alignments, longest span of source words is searched which are all generated from the target. In addition, a set of constraints are imposed on both source and target fragments with respect to stopwords, length of fragments and number of unaligned words on target side.

But the conditional model is asymmetric and has a lot of free parameters to be tuned. Hence, this approach is extended to a joint conditional model. The joint model chooses between three options while generating: source only fragment, target only fragment or bilingual fragment. Since conditional models generally outperforms joint models, $p(e, f)$ is decomposed as $p(e, f) = p(e)p(f|e) = p(f)p(e|f)$ and the minimum of two generative models is used as an estimate for the joint probability. Then the generative framework proceeds by first predict the number of fragments. Then for each fragment, predict the number of source and target words generated by the fragment and then finally generate the source and target words in each fragment. Parallel fragments are finally extracted by searching for the most likely sequence of fragments.

3.6 Text entailment

(Pal et al., 2014) uses a combination of text entailment and an existing machine translation system to extract parallel fragments of text from comparable documents. They translate the source sentence into target sentence and compute a text entailment score between the target sentence and the translated source sentence in both directions. If the text entailment score is higher than 50%, then they are considered for further processing.

Text entailment module

The text entailment module uses a combination of various lexical, syntactic and semantic features to detect text entailment in system. Lexical features include unigram match, bigram match, longest common sequence, skipgram, stemming and named entity matching. The syntactic module compares the dependency relations in both hypotheses and text. The system creates a parse tree of both sentences and compares the two to judge their similarity. In order to measure the semantic similarity between two sentences, they are converted into UNL form and then the UNL form is compared and assigned a score. The extracted features are then finally trained using a SVM.

Extraction of parallel fragments

To extract parallel fragments from the sentences, first all the sentences that have text-entailment score above 50% are grouped together into a cluster. All the source and target sentences within the same cluster are then intersected. To extract the phrases from these sentence pairs, a seed corpus of parallel fragments are used. New fragments are extracted using these fragments and added back to the seed fragments. These new fragments are then used again to extract new fragments.

3.7 Summary

This chapter gave various approaches that are used to extract parallel phrases from comparable corpora. These approaches use various ideas like log likelihood ratio, chunking, text entailment to extract parallel phrases. Some of these approaches also make use of alignment between words in sentence either directly by extracting phrases from them or indirectly by using alignment features in a classifier.

Chapter 4

Bilingual lexicon extraction

Another application of comparable corpora is in the domain of extracting bilingual lexicons. Bilingual lexicons play an important role in many NLP applications. Hence, we give an overview of some of the approaches used for same. We divide the discussion based on two broad types of approaches which are co-occurrence based and topic based models.

4.1 Context based approaches

The key idea behind this approach is based on distributional hypotheses that terms and its translations tend to occur in similar lexical context across languages. Various methods have been proposed that try to use this observation. Below, we first describe a standard approach making use of this. Next, we try to describe a method, that finds an analogy between finding words with similar contexts and document retrieval technique.

4.1.1 Standard approach

In one of the most standard approach that have been used extensively, there are basically four steps involved: build context vectors, map context vectors to a common space, find similarity between context vectors, extract word translations.

Build context vectors

In this step, one obtains all the words that occur in the same context as the candidate word. The context of word may be defined at multiple levels of granularity. The context word could be within a window of words, or words within the same sentence or same paragraph. Next, for each context word, a suitable measure is obtained which defines the level of associativity between the context word and the candidate word. In a typical setup, one usually obtains the counts of co-occurence of the context words and candidate word and then normalise it by a Mutual Information score or Log Likelihood. Another way to obtain a score is to use TFIDF based approach, where TF is the number of times the context word and candidate word co-occurs and IDF of a context word w_i is defined as follows:

$$IDF_i = log\frac{maxn}{n_i} + 1$$

where $maxn$ is the maximum frequency of any word in the corpus and n_i is the total occurences of word w_i.

The next question regarding the choice of context word is which word should be chosen as a context word. Typically, content words are chosen as context words which have known translation in other language. Further, one may choose to use such context words which occur very frequently in the corpus.

Using any of described techniques, a context vector is then constructed for both source word and target word.

Mapping vectors to a common domain

Next, it is observed that source vector and target vector are built in different language with different dimensions. Thus, they cannot be directly compared. In order that the two vectors be directly comparable, we translate all the words in the source vector into the target language using a seed vector. Thus, here we see that the choice of context vectors is dependent on the seed dictionary that is available.

Find similarity

Once, we get vectors similiarity between the two vectors is computed using various similarity metrics, cosine similarity being the most common one. Let the two context vectors $W_1 = (w_{11}, w_{21}, \ldots, w_{t1})$ and $W_2 = (w_{12}, w_{22}, \ldots, w_{t2})$. The cosine similarity between them is then computed as shown below where t is the dimension of the context vector:

$$Sim(W_1, W_2) = \frac{\sum_{i=0}^{t}(w_{i1} * w_{i2})}{\sqrt{\sum_{i=0}^{t} w_{i1}^2 * \sum_{i=0}^{t} w_{i2}^2}}$$

Extracting word translations

For each source word, a similarity is computed for each candidate target word, and then the word pair with highest similarity is considered as translations of each other. Although, sometimes instead of selecting the pair with highest score, n-top target words are output as potential translations to improve the precision of extraction. Word pairs that have similarity above a threshold can be considered to be translations of each other.

4.1.2 Analogy to Document retrieval

(Shao and Ng, 2004) in their approach view the problem of finding whether two words have a similar context as a document retrieval problem. In their approach, the context of a word f in language1 $L1$ is viewed as a query and the context of each candidate translation of word e in target language $L2$ is viewed as the set of documents. With such a formulation, they then try to find the most similar document(here, context of most likely translation of $word1$) by posing the query(context of $word1$)

In their approach, they use an approach taken from the field of information retrieval called language modelling approach. In this approach, a language model is derived for each document D. Then, the probability of generating the query Q according to that language model, $P(Q|D)$, is estimated. The document with the highest $P(Q|D)$ is the one that best matches the query.

Estimation of $P(Q|D)$

The computation of $P(Q|D)$ is inspired by the work done by (Ng, 1999). Here, they first represent the document D as a multinomial distribution of terms and assume that query Q is generated by this model:

$$P(Q|D) = \frac{n!}{\prod_t c_t!} \prod_t P(t|D)^{c_t}$$

where t is a term in the corpus, c_t is the number of times term t occurs in the query Q, $n = \sum_t c_t$ is the total number of terms in query Q.

It was then observed that first part in the above equation involving fraction can be ignored as it depends only on the query and is same for all documents.

In the current scenario, the query Q is the $C(f)$ and the document is the $C(e)$ where we define $C(word)$ as the context of the word in its respective language. So, our modified problem is to compute $P(C(f)|C(e))$ for each word e and find the word e with maximum $P(C(f)|C(e))$. In the revised scenario, $P(C(f)|C(e))$ is computed as follows:

$$P(C(f)|C(e)) = \prod_{t_f \in C(f)} P(t_f|T_f(C(e)))^{q(t_f)}$$

where, term t_f is a word in language $L1$, $q(t_f)$ is the number of occurrences of t_f in $C(f)$, $T_f(C(e))$ is the bag of words of language $L1$ obtained by translating the words in $C(e)$, as determined by a bilingual dictionary. Then, probabilities are estimated by using backoff and linear interpolation as follows:

$$P(t_f|T_f(C(e))) = \alpha P_{ml}(t_f|T_f(C(e))) + (1 - \alpha)P_{ml}(t_c)$$

$$P_{ml}(t_f|T_f(C(e))) = \frac{d_{T_f(C(e))}(t_f)}{\sum_{t \in T_f(C(e))} d_{T_f(C(e))}(t_f)}$$

where $P_{ml}(\bullet)$ are the maximum likelihood estimates, $d_{T_f(C(e))}(t_f)$ is the number of occurrences of the term t_f in $T_f(C(e))$, and $P_{ml}(t_f)$ is estimated similarly by counting the occurrences of t_f aftre translating the corpus in language $L2$ into $L1$.

Finally, the word e with highest probability is output as the translation of a word f.

4.2 Classifier based approach

4.2.1 Random forest classifier

(Georgios et al., 2014b) propose the use of a random forest classifier to extract bilingual lexicons from a comparable corpora. In their proposed approach, they represent the source and target terms as a feature vector where each dimension corresponds to a unique character n-gram. The value of the dimension is set to be 1 or 0 depending on whether the character ngram is contained in the term. The feature vectors used finally consists of q source ngrams and q target ngrams.

The key component that helps the system in learning character gram mappings is the decision tree. A node in the tree is equivalent to a unique ngram. The nodes are linked through the branches of the trees and therefore the two sub-spaces of q source and q target character grams are combined. Then each decision tree in the forest is constructed as follows: every node is split by considering $|\psi|$ random n-gram features of the initial feature set ω, and a decision tree is fully grown. This process is repeated $|\tau|$ times and constructs $|\tau|$ decision trees.

The classification margin that we use to rank the candidate translations is calculated by simply subtracting the average number of trees predicting that the input terms are not translations from the average number of decision trees predicting that the terms are mutual translations. A larger classification margin means that more decision trees in the forest classify an instance as a translation pair.

4.2.2 Discriminative classifier

(Irvine and Callison-Burch, 2013) propose use of discriminative classifier to extract bilingual lexicons. The feature set they used cover a wide range of features including temporal, contextual, topic, orthographic, and frequency similarity between a candidate translation pair. The model is trained on positive instances extracted from an aligned parallel corpora and negative instances obtained by randomly mapping incorrect words. During extraction of word translations, each unknown source word is mapped with all target words in the corpus and then they are ranked by their classification scores thereby extracting the best one.

4.3 Combined string and context similarity

(Georgios et al., 2014a) propose a method to extract word translations combining both string level similarity and context similarity. They observe that bilingual terms are often formed by mapping sub-lexical units across languages. Another observation they make is that the terms tend to occur in similar context across languages. Taking into account these observations, they train a logistic regression classifier, for learning a string similarity measure of term translations. While they use an existing context based approach according to the second observation. The feature set used for training logistic classifier is similar to (Georgios et al., 2014b) where they consider the presence or absence of a ngram as a feature. In addition, they also consider second order features where each feature is a tuple of ngrams. Thus each feature is 1 or 0 depending on whether the source and target term contains all the ngrams present in the tuple. Thus, if there are p source ngrams and q target ngrams, there are in all $p \times q$ possible second order features.

Finally they combine the results obtained by combining the outputs of both string based model and context based model.

4.4 Topic models

These models are based on the idea that two words are likely to be translation of each other if they are present in documents on a same topic and not present in other topics. Based on this observation, (Vulić et al., 2011) use an extension of the standard LDA to a bilingual environment which is BiLDA. This is then applied on a set of document aligned comparable corpora. In its formulation, BiLDA uses a single parameter θ that models the topic distribution. The topics for each document are sampled from this θ following which words are sampled from them in conjugation with distributions of source vocabulary ϕ and target vocabulary ψ. The parameters are then trained using a Gibbs sampler and a set of word topic distributions θ and ψ are obtained. At the end of this, a shared set of topics along with language-specific distributions of words over topics is obtained. Then, bilingual lexicons are extracted by measuring their associations through various measures like KL divergence, cue and term frequency and inverse topic frequency(TF-ITF).

4.5 Summary

In this chapter, we discussed various approaches that have been used previously to extract word translations from comparable corpora. They include co-occurrence based models, topic based models and context based models. Further, we also described classifier based approaches which make use of the above mentioned features.

Bibliography

Abdul Rauf, S. and Schwenk, H. (2009). On the use of comparable corpora to improve smt performance. pages 16–23.

Fung, P. and Cheung, P. (2004). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and e. In *EMNLP*, pages 57–63.

Fung, P. and McKeown, K. (1997). Finding terminology translations from non-parallel corpora. *Proceedings of the 5th Annual Workshop on Very Large Corpora*, pages 192–202.

Fung, P. and Yee, L. Y. (1998). An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 414–420. Association for Computational Linguistics.

Gaizauskas, A. A. Y. F. R. (2012). Automatic bilingual phrase extraction from comparable corpora. pages 23–32.

Georgios, K., Ioannis, K., Jun'ichi, T., and Sophia, A. (2014a). Combining string and context similarity for bilingual term extraction from comparable corpora. In *EMNLP*, pages 1701–1712.

Georgios, K., Ioannis, K., Jun'ichi, T., and Sophia, A. (2014b). Using a random forest classifier to compile bilingual dictionaries of technical terms from comparable corpora. In *Proc. of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 111–116.

Irvine, A. and Callison-Burch, C. (2013). Supervised bilingual lexicon induction with multiple monolingual signals. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, NAACL '13.

Jakob, U., Jay, M. P., Ashok, C. P., and Moshe, D. (2010). Large scale parallel document mining for machine translation. pages 1101–1109.

Jason, R. and Daniel, M. (2012). Automatic parallel fragment extraction from noisy data. In *NAACL HLT '12 Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, pages 538–542, Stroudsburg, PA, USA. Association for Computational Linguistics.

Moore, R. C. (2004). Improving ibm word-alignment model 1. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Munteanu, D. S. and Marcu, D. (2002). Processing comparable corpora with bilingual suffix trees. pages 289–295.

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Munteanu, D. S. and Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 81–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ng, K. (1999). A maximum likelihood ratio information retrieval model.

Och, F. J. and Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 1086–1090, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pal, S., Pakray, P., and Naskar, K. S. (2014). *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, chapter Automatic Building and Using Parallel Resources for SMT from Comparable Corpora, pages 48–57. Association for Computational Linguistics.

Philips, L. (1990). Hanging on the metaphone. *Computer Language Magazine*, 7(12):39–44. Accessible at `http://www.cuj.com/documents/s=8038/cuj0006philips/`.

Quirk, C., Udupa, R., and Menezes, A. (2007). Generative models of noisy translations with applications to parallel fragment extraction. In *In Proceedings of MT Summit XI, European Association for Machine Translation.*

Rajdeep, G., Santanu, P., and Sivaji, B. (2013). Improving mt system using extracted parallel fragments of text from comparable corpora. In *6th Workshop of Building and Using Comparable Corpora (BUCC)*, ACL, pages 69–76. Association for Computational Linguistics.

Resnik, P. and Smith, N. A. (2003). The web as a parallel corpus. *Comput. Linguist.*, 29(3):349–380.

Shao, L. and Ng, H. T. (2004). Mining new word translations from comparable corpora. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Smith, J. R., Quirk, C., and Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 403–411. Association for Computational Linguistics.

Stephan, V., Hermann, N., and Christoph, T. (1996). Hmm-based word alignment in statistical translation. In *COLING '96 Proceedings of the 16th conference on Computational linguistics - Volume 2*, pages 836–841, Stroudsburg, PA, USA. Association for Computational Linguistics.

Toutanova, K. and Galley, M. (2011). Why initialization matters for ibm model 1: Multiple optima and non-strict convexity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 461–466, Stroudsburg, PA, USA. Association for Computational Linguistics.

Vulić, I., De Smet, W., and Moens, M.-F. (2011). Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 479–484, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403.

Wu, D. and Fung, P. (2005). Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, IJCNLP'05, pages 257–268, Berlin, Heidelberg. Springer-Verlag.