Extraction of Parallel Corpora from Comparable Corpora

Survey Report

by

Rucha C. Kulkarni

under the guidance of

Prof. Pushpak Bhattacharyya

Department of Computer Science & Engineering,

Indian Institute of Technology, Bombay

Acknowledgement

I express my sincere and heartfelt gratitude towards my guide, Prof. Pushpak Bhattacharya, for all the valuable guidance and support he has given me in this literature survey. I also thank the entire Machine Translation group at CFILT, IIT Bombay, for the insightful discussions and their valuable suggestions in the weekly meetings.

Rucha Kulkarni
(Department of Computer Science and Engineering)

Abstract

The size and quality of the parallel corpus used for training, greatly impacts the quality of translation of an SMT system. But, there are very few sources of parallel corpora for many language pairs. This is a major hurdle in the development of good SMT systems. To alleviate this problem, comparable or non-parallel corpora, which are largely available, can be exploited to extract parallel data. We study the recent work done in this area, and explore various approaches for extraction of parallel sentences, parallel fragments of sentences and bilingual lexicons from comparable corpora.

Contents

List of Figures

Chapter 1

Introduction

Machine Translation (MT) is the task of automatically converting one natural language text into another while preserving the meaning of the input text, and producing fluent text in the output language. There are various approaches to MT:

- **Rule Based MT:** A rule-base for translation between source and target languages is created and the target sentences are generated according to these rules. This rule-base consists of rules of grammar for both languages and a dictionary for word translations.

- **Interlingua Based:** Translation is done by using an intermediate representation of the text (interlingua) between source and target languages.

- **Transfer Based:** The types of transfer based approaches are — syntactic transfer and semantic transfer. In the transfer approach, a syntactic (or semantic) structure of the source sentence is built and it is transformed into a syntactic (or semantic) structure of the target sentence. The target sentence is generated from this structure. Semantic methods use a richer semantic representation than parse trees.

- **Statistical MT:** Statistical MT learns the statistics over parallel corpora and estimates parameters of a probabilistic model for translation.

The Vauquois' Triangle in Figure 1.1 shows the various strategies of machine translation. As we go up in this triangle, understanding of the source and target language required

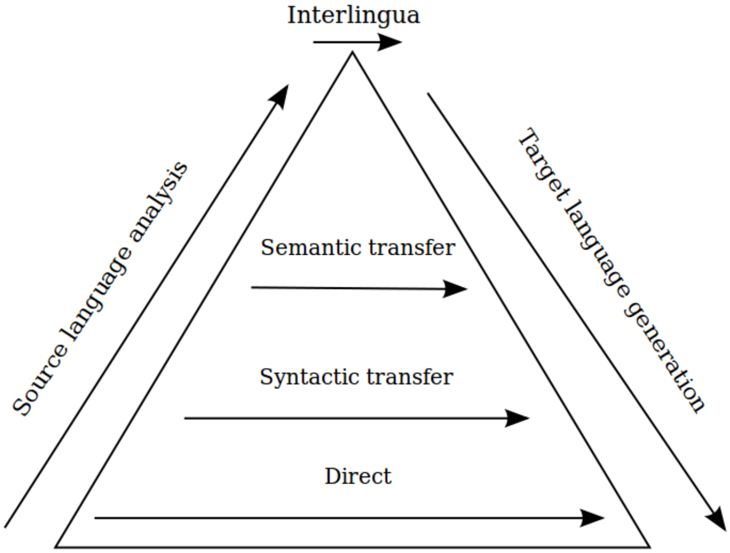


Figure 1.1: Vauquois' Triangle

for the translation, increases (Manning and Schütze, 1999).

1.1 Statistical Machine Translation

Language is so rich and diverse that it can never be fully analyzed and written into a set of rules, which can be encoded in a computer program (Koehn, 2010). Hence, the statistical approach to MT gained momentum. In SMT, the goal is to develop a machine that discovers the rules of translation automatically from a large corpus of bilingual text. Thus, the rules of translation are learned by the system from the statistics over the data.

Translation is done according to the probability distribution p(f|e) over the documents. There are various approaches for modeling this probability distribution. Initially, the statistical translation models were word-based *i.e.*, a word was the fundamental unit of translation. Such systems used IBM models 1 to 5 that are based on the IBM Hidden Markov Model (Vogel et al., 1996) and model 6 (Och and Ney, 2003). Later, phrase-based models came into being . There has also been work on syntax-based and hierarchical phrase-based machine translation (Chiang, 2005).

1.1.1 Importance of large amounts parallel corpora for SMT

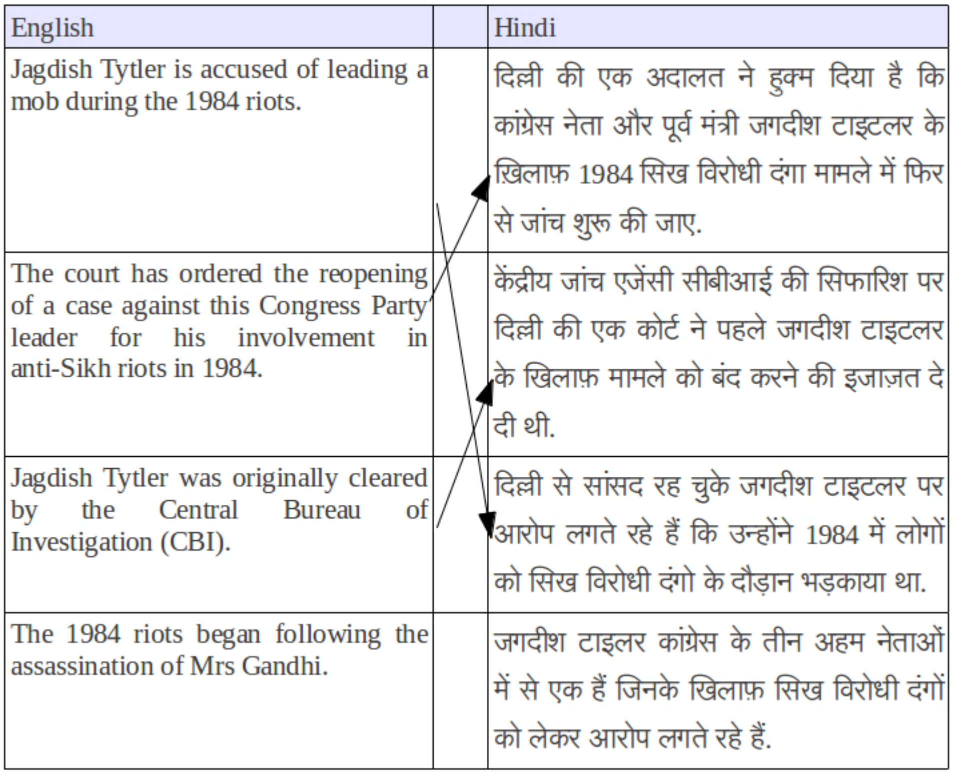
A machine translation system that is based on the probabilistic translation models (Brown et al., 1993), are generally trained using parallel corpora. A parallel corpus is a sentence aligned pair of documents in which each pair of aligned sentences are translations of each other. During training, the translation model of the SMT system learns the statistics over the training data provided to it and estimates its parameters accordingly. The translations that are produced are very much dependent on the parallel corpus used for training. Larger the size of the training corpus, better is the parameter estimation and thus, better is the translation quality.

But, one of the major challenges faced by SMT is that of scarce availability of parallel corpora. There are some language pairs like English-French or English-Spanish, for which huge set of parallel corpora are readily available. But for many other language pairs, there is scarcity of parallel corpora.

Creation of a large parallel corpus manually, can be very costly in terms of efforts and man hours. So, we need some ways to automatically and efficiently create parallel corpora. Comparable corpora and non-parallel corpora are largely available for all language pairs. The next section introduces comparable corpora and its various sources.

1.2 Comparable Corpora

A noisy parallel or comparable corpus consists of bilingual documents that are not sentence aligned, but are rough translations of each other. In fact, in a noisy parallel corpus, the documents can possibly contain many parallel sentences in a roughly same order (Smith et al., 2010). But, in a comparable corpus, the sentences are not really translations of each other but convey almost the same information, and hence, may contain some parallel sentences. Such comparable corpora can be exploited to find parallel sentences or parallel phrases. Examples of such comparable corpora are multilingual news feeds provided by news agencies like Agence France Presse, Xinhua News, Reuters, CNN, BBC, etc (Munteanu and Marcu, 2005). Also, comparable corpora could contain documents that are not even rough translations of each other but are only topic-aligned.



English		Hindi
Jagdish Tytler is accused of leading a mob during the 1984 riots.		दिल्ली की एक अदालत ने हुक्म दिया है कि कांग्रेस नेता और पूर्व मंत्री जगदीश टाइटलर के ख़िलाफ़ 1984 सिख विरोधी दंगा मामले में फिर से जांच शुरू की जाए.
The court has ordered the reopening of a case against this Congress Party leader for his involvement in anti-Sikh riots in 1984.		केंद्रीय जांच एजेंसी सीबीआई की सिफारिश पर दिल्ली की एक कोर्ट ने पहले जगदीश टाइटलर के खिलाफ़ मामले को बंद करने की इजाज़त दे दी थी.
Jagdish Tytler was originally cleared by the Central Bureau of Investigation (CBI).		दिल्ली से सांसद रह चुके जगदीश टाइटलर पर आरोप लगते रहे हैं कि उन्होंने 1984 में लोगों को सिख विरोधी दंगो के दौड़ान भड़काया था.
The 1984 riots began following the assassination of Mrs Gandhi.		जगदीश टाइलर कांग्रेस के तीन अहम नेताओं में से एक हैं जिनके खिलाफ़ सिख विरोधी दंगों को लेकर आरोप लगते रहे हैं.

Figure 1.2: Example of Comparable Corpus

Consider the set of sentences shown in Figure 1.2. It is a small example of a comparable corpus. These are sentences from the newspaper articles reporting the same incident on BBC India website English [1] and Hindi [2]. The arrows point to the corresponding comparable sentence that provides the same information. As can be observed, alignment of parallel sentences such as these is clearly, a non-trivial task, as there may be many lexical and structural differences in the two sentences, inspite of them giving the same information.

1.2.1 Wikipedia

Wikipedia is a huge collection of articles on a large variety of topics and various languages. So, it is rich in information from various domains and that too, in many different languages.

Interwiki Links

Articles on the same topic in different languages in Wikipedia are connected through *interwiki links*. These links are annotated by users. Thus, document alignment for multilingual documents on similar topics is already provided in Wikipedia. These aligned documents, can be directly given to the sentence extraction step.

Markup

Wikipedia's markup can be very useful in providing numerous cues for parallel sentence extraction. For example, text in a typical Wikipedia article contains hyperlinks that point to other articles. If the hyperlinks in a bilingual sentence pair match, then that sentence pair can be said to be parallel. Hyperlinks match if the articles they point to, are connected by interwiki links.

[1] http://www.bbc.co.uk/news/world-asia-india-22093997

[2] http://www.bbc.co.uk/hindi/india/2013/04/130410_tytler_re-investigation_fma.shtml

Image Captions

Images in Wikipedia are stored centrally across all articles in all languages. So, if the same image occurs in an aligned pair of articles, then its caption in both those articles can be used as parallel sentences.

Lists and Section Headings

Lists and section headings can be useful in finding similar content in the articles.

Thus, there are many sources of useful information in Wikipedia which can be exploited for our purpose. But, the aligned article pairs of Wikipedia may be of three types. First, they may be translations of each other, in which case, these are almost parallel. Second, they may be translations but edited independently. So, they can be called noisy parallel. And third is that, the articles may be written by different authors. So, inspite of being topic aligned, they may contain no parallel sentences. Hence, Wikipedia ranges from being a collection of noisy parallel to comparable article pairs (Smith et al., 2010).

1.2.2 Quasi-Comparable Corpora

A quasi-comparable corpus (Fung and Cheung, 2004b) contains non-parallel and non-aligned bilingual documents. These documents may be on the same topic or may be of very different topics. A good example of quasi-comparable corpus is the TDT3 Corpus, which consists of transcriptions of radio broadcasts and TV news reports. In such a corpus, a small number of the bilingual sentences are translations of each other, while some others are bilingual paraphrases.

1.2.3 The Internet Archive

The Internet Archive (Resnik and Smith, 2003) is a non-profit organization that is attempting to archive the entire Web. It preserves content in web pages and makes them freely and publicly available through a Wayback Machine Web Interface [3]. The data of the Archive is freely accessible by signing up for an account on their cluster. The data are stored on disk drives of more than 300 machines, each running on UNIX.

Properties of the Archive

In 2003, it contained around 120TB of data (a conservative estimate) and grows at the rate of 8TB per month.

1. **Architecture of the Archive:**

   - There are many aspects of the architecture of the Archive that make it very feasible to rapidly develop tools for processing it.
   - All data are stored in the form of compressed files rather than in a database.
   - The data relevant for our extraction purposes are organized as archive files (arcfiles) containing the stored page, the index files containing tuples of the form <URL, timestamp, arcfile, offset,...>
   - There are a number of tools available to process the Archive and extract individual pages from the archive files.
   - The Archive is divided on a cluster of computers. So, it facilitates the use of cluster computing and makes it easy to write small UNIX scripts for parallel processing across the machines.
   - Parallel processing makes the extraction process remarkably fast.

[3] www.archive.org/

2. **Challenges in mining the Archive:**

   - The Archive is a temporal database but, it is not stored in temporal order. So, a document and its translation may be kept on different machines and a global merge is required to find them.
   - All data is stored in compressed format. So, extracting some text even for the purpose of inspection is a very expensive task involving text decompression.
   - The Archive is so big in size that, all the processing that we do on the data must be done using algorithms of least computational complexity.

1.3 Summary

In this introduction chapter, we discussed the various approaches of Machine Translation (MT), and saw the importance of parallel corpora for Statistical Machine Translation (SMT). We discussed some huge, and freely available sources of comparable corpora, and their features that help us find topic aligned documents in them.

1.4 Outline of the Report

Chapter 2 gives a general architecture of a parallel corpora extraction system. Chapter 3 discusses approaches for document alignment. In Chapter 4, the existing work in extracting parallel sentences from non-parallel or comparable corpora using various approaches has been discussed. In Chapter 5, we discuss some approaches for extraction of parallel fragments from comparable corpora. In chapter 6, we survey approaches for bilingual lexicon extraction from comparable corpora.

Chapter 2

Parallel Corpora Extraction System: General Architecture

The extraction system or the pipeline is made up various stages as shown in Figure 2.1. Firstly, potential sources of texts that could yield parallel sentences must be looked for. These could be any known noisy parallel, comparable, or quasi-comparable corpora that were seen in Chapter 1. We can also extract parallel data from web sources, as has been discussed.

A parallel corpora extraction system generally consists of the following:

1. **Resources**

   - Comparable or non-parallel corpus: This is the source for parallel sentence extraction. It consists of documents in the source and target language.
   - Bilingual Lexicon: Used in the Document Alignment and the Sentence Selection stages.
   - Seed Parallel Corpus: A small parallel corpus used to train the Sentence Selection module.

2. **Document Alignment Module**

   After identifying the potential source, the next step is to find similar document pairs by document alignment (topic alignment). This is an important step because of the “find-topic-extract-sentence” principle (Fung and Cheung, 2004a). It says that parallel sentences are most likely to be found in documents on similar topics.

3. **Sentence Selection Module**

   After document alignment, parallel sentences within the similar document pairs are extracted. This is done using various Maximum Entropy based models that make use of feature functions (Munteanu and Marcu, 2005).

4. **Bootstrapping**

   Another principle - “find-one-get-more”, states that, if one parallel sentence pair is found in a document pair, then it must contain more parallel sentence pairs (Fung and Cheung, 2004a). To take advantage of such phenomena, bootstrapping (not shown in 2.1) can be employed.

Each of these stages has been discussed in detail in the following chapters.

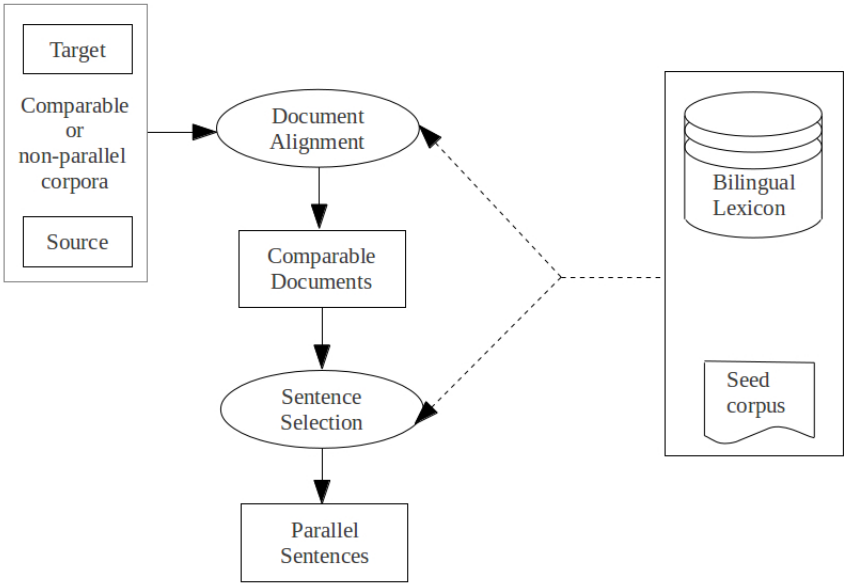


Figure 2.1: General Architecture of Parallel Sentence Extraction System

Summary

The general architecture of a parallel sentence extraction system was seen. The subsequent chapters discuss various techniques for implementing each of the steps mentioned in this chapter.

Chapter 3

Document Alignment Techniques

A comparable or non-parallel corpus is likely to be huge. It is not possible to examine every sentence pair in the entire corpus. So, we focus our attention on only those sentence pairs that belong to documents having similar or overlapping content. Hence, we need to find comparable or similar documents from the set of all documents. Following are some of the techniques that can be employed.

- TFIDF Retrieval
- Cosine Similarity
- Topic Alignment
- Content Based Alignment

We will discuss each of these in detail.

3.1 TFIDF Retrival

In this method (Munteanu and Marcu, 2005), for each foreign lanuage document, we try to select a list of English documents that are likely to contain sentences which are parallel to those in the given foreign document.

If we find the best matching English document for a foreign language document, then the number of candidate sentence pairs to be given for classification is too small, and we may miss out on a large number of good candidates. So, we try to find a set of matching English documents instead of finding a single best-match, since this delegates the load of the extraction pipeline towards the next step - of actual sentence alignment and extraction, which is more robust and reliable.

*TF-IDF = Term frequency * Inverse Document Frequency*

This is a metric to show how important the given word is to a document. TFIDF of a document increases if a word in more frequent in that document, but reduces if the words also appears frequently in the entire corpus. Thus, it helps to ignore words that are inherently common, as they do not contribute much towards the content of the document.

TFIDF is used to compute a ranking function to rank documents according to their relevance to a given query of words. Many IR and text mining tools make use of a scoring or ranking functions based on TFIDF and its variations. In this technique, we can make use of any such IR toolkit. We will also make use of a probabilistic bilingual dictionary. Following are the steps of document matching:

- Index all English documents in a database based on the words it contains.
- For a given foreign language document, we take the top 5 translations of each of its words according to the dictionary and construct an English language query.
- This query is used to run a TF-IDF retrieval of English documents against the database.
- Upto 20 matching English documents may be paired for the given foreign language document.
- This process is followed for each foreign document.

3.2 Cosine Similarity

Cosine similarity is a measure of similarity in two documents. The documents should be represented as a vector of the words they contain. Then the cosine similarity is nothing but the cos of the angle between the representative vectors *i.e.* Dot product of the vectors.

Alternattely, TFIDF and cosine similarity can be combined to give improved results by representing each document by a TFIDF vector. Thus, factors like length of the document will not affect the similarity score, giving more accurate results.

3.3 Topic Alignment

In noisy-parallel or comparable corpora, the documents are mostly on same or relevant topics. Thus, these are said to be topic-aligned. Many recent works are based on the find-topic-extract-sentence principle (Fung and Cheung, 2004a) which says that parallel sentences exist in document pairs with high similarity.

But, in highly non-parallel corpora, documents may or may not be on the same topic. Thus, generally, they are not topic-aligned. Hence, we need to find similar *i.e.* in-topic documents using methods like word overlaps, similarity scores, TFIDF, and cosine similarity.

3.4 Content Based Alignment

This method uses a translational similarity score based on a word-to-word translation lexicon (Resnik and Smith, 2003).

Link: It is defined as a pair (x,y) where x is a word in foreign language and y is a word in English language. Either x or y, but not both can be NULL, indicating that the translation on the opposite side is not known.

We use a generative, symmetric model based on a bilingual dictionary that gives a probability distribution 'p' over all possible link types in the corpus. This model does not take word order into account.

Let us consider two documents X (in foreign language) and Y (in English). We, need to find the most probable link sequence(unordered) that can account for both these documents according to our model.

$$\mathit{\Psi Pr(link\text{-}sequence)} = \Pi_l \; \mathit{Pr(x,y)}$$
$$\mathit{where,\ l = (x,y)}$$

So, finding the best set of links over X and Y is a problem of finding the Maximum Weighted Bipartite Matching(MWBM). In our case, the weights of the links are the log of Pr(x,y). So, maximum sum of the weights gives most probable link sequence(unordered).

Tsim: this is defined to be a cross-language similarity score between two documents based on the model defined earlier. This similarity score should be high when many of link tokens in the best link sequence do not have NULL on any side. Also, it should be normalized for text length.

$$\mathit{Tsim} = \frac{\Sigma\mathit{(log\ Pr\ (two\text{-}word\ links\ in\ best\ matching))}}{\Sigma\mathit{(log\ Pr\ (all\ links\ in\ best\ matching))}}$$

A drawback of this score is that a lot of probability mass in the distribution 'p' tends to go to frequent words, which are relatively uninformative for the problem of finding whether two documents contain content translations of each other.

So, we can assume an equiprobability model. Here, all links have equal probability. Thus, the MWBM problem is reduced to a Maximum Cardinality Bipartite Matching (MCBM) problem under the equiprobability assumption. So, best link sequence is the one having maximum number of link tokens. Then,

$$Tsim = \frac{number\ two\text{-}word\ links\ in\ best\ matching}{number\ of\ links\ in\ best\ matching}$$

The document pairs with highest Tsim score can be considered as relevant or similar documents.

We can combine multiple sources of word-level translation information in the model like dictionaries, word-to-word translation model as described above and cognates.

Summary

In this chapter, we saw various techniques of document alignment like TFIDF Retrieval, Cosine Similarity, Topic Alignment and Content Based Alignment. The next chapter discusses sentence alignment techniques.

Chapter 4

Extraction of Parallel Sentences

Extracting parallel sentences from comparable corpora is a vast area of research in itself. It has continued to remain a challenging task even after more than a decade. Some standard and most commonly used techniques are explained in this section.

Most of the techniques are implemented on a set of bilingual document pairs that are topic-aligned, i.e., each pair of documents have content about roughly similar topics, so that they tend to convey a lot of overlapping information. This improves our chances of finding good parallel sentences while reducing the search space, too.

Wikipedia is a good example of comparable corpus that readily provides such topic-aligned pair of documents through its *"interwiki"* links. But when we do not have such a paired set of documents, techniques discussed in the previous chapter can be used to align documents. Such pairs of documents, then, can be used to find parallel sentences. But, we need a reliable way of finding parallel sentence pairs such document pairs. Following are some techniques that can be used for classifying parallel sentence pairs from all sentence the pairs in the aligned set of documents.

4.1 Classification

A Maximum Entropy classifier was used for parallel sentence extraction by Munteanu and Marcu (2005). The model is a log linear combination of various feature functions:

$$P(ci|sp) = \tfrac{1}{Z(sp)} \cdot \prod_{j=1}^{k} \lambda_j^{f_{ij}(c,sp)}$$

where c_i is the class, c_0 = parallel and c_1 = non-parallel

Z(sp) is the normalization factor

f_{ij} are the feature functions.

Since, there are only two classes - parallel and non-parallel, this is a binary classifier. It takes as input a candidate sentence pair and according to the model, classifies it as either parallel or non-parallel with some probabilistic score (confidence). We can further improve precision by taking only those pairs which are classified as parallel with a high probability (say 0.7 or greater).

The training set for this classifier can be a small parallel corpus of about 5K sentences. Then, all possible sentence pairs (Cartesian product) are generated. So, 5000^2 training instances will be generated out of which, 5000 are parallel and remaining are not parallel. In any given training data set of 'n' sentence pairs, there are usually O(n) positive (parallel) examples and O(n^2) negative (non-parallel) examples. A binary classifier trained on such a training instance suffers from a class imbalance problem. As a result, it tends to predict the majority class *i.e.*, classifies most pairs as non-parallel.

One heuristic solution to this problem is to downsample *i.e.*, randomly remove negative instances from the training set to achieve balance. Also, we can first pass all the generated pairs through the word overlap filter, and use for training, only those pairs that are not discarded by the filter. This can reduce class imbalance since the pairs that will be discarded are least likely to be parallel.

4.2 Maximum Entropy Based Ranking Model

To avoid the class imbalance problems of a binary classifier, a Maximum Entropy based Ranking Model was used by Smith et al. (2010). The ME model and training set used are same as those of the classifier described above but the formulation of the problem is different.

4.2.1 Formulation

For each source language sentence, we select a sentence in the target language that is the most parallel to it. This is selected based on the probability score of the ME model used. Thus, highest probability sentence from target is chosen as parallel to the given source sentence. If no such target sentence is found, then the source sentence is aligned to NULL.

This formulation of the problem reduces the class imbalance problem of the binary classifier.

4.3 Sentence Similarity

Fung and Cheung (2004a) use sentence similarity technique, where, each sentence is represented as a word vector. Then, pairwise sentence similarity is calculated for all possible sentence pairs in the aligned document pairs. A minimum threshold is set and sentence pairs yielding a similarity score beyond this threshold are considered to be parallel. Similarity score may be computed using TF-IDF (in this case, document is a sentence) and cosine similarity.

4.4 Conditional Random Fields

Smith et al. (2010) used a first order linear chain Conditional Random Field (CRF) for aligning parallel sentences within a pair of aligned documents. This model corresponds to the discriminative CRF-based word alignment model described by Blunsom and Cohn (2006). This alignment model is applied for aligning sentences within aligned documents instead of words within parallel sentences.

4.4.1 Discriminative Word Alignment with Conditional Random Fields

Blunsom and Cohn (2006) describe a CRF Sequence model estimated on a small supervised training set of word aligned parallel sentences. This model allows for a globally optimal training and decoding and uses a graphical structure similar to the directed hidden Markov model (HMM) from GIZA++ (Och and Ney, 2003). It models many-to-one alignments where each source word is aligned with zero or one target word, so each target word can be aligned to multiple source words. Many-to-many alignments are obtained by superimposing the predicted alignments in both translation directions.

1. **Formulation:**

   Each source word is labeled with the index of the target word to which it is aligned, or *'null'*. The joint probability density of alignment, conditioned on the source and target sides is

   $$p_{\wedge}(a|e,f) = \frac{exp \sum_t \sum_k \lambda_k h_k(t, a_(t-1), a_t, e, f)}{Z_{\wedge}(e,f)} \tag{4.1}$$

   where a = alignment vector containing target indexes,

   e = source sentence,

   f = target sentence,

   t ranges over target side indexes,

   k ranges over model features

   $\wedge = \lambda_k$ the model parameters (weights for their corresponding features)

   h_k = real valued feature functions over source and target sentences

   $$Z_{\wedge} = \sum_a exp \sum_t \sum_k \lambda_k h_k(t, a_(t-1), a_t, e, f) \tag{4.2}$$

   $Z_{\wedge}$ = partition function to normalize the distribution in equation 4.1

2. **Feature Set:**

   The feature set includes two main types of feature functions: features defined on a candidate aligned word pair, and Markov features defined on the alignment sequence predicted by the model.

4.4.2 CRF Based Parallel Sentence Extraction

The parallel sentence extractor based on the CRF model described above uses a subset of the rich set of features, described in the next section.

4.5 Features Functions

The classification based approach, as well as CRF based approach uses a lot of feature functions that they help to distinguish between parallel and non-parallel sentence pairs. For this purpose features based on word alignments, translation probability, distortion, context similarity, etc are used (Munteanu and Marcu, 2005).

4.5.1 General Features

1. **Sentence Length**

   The length of the candidate sentences should be roughly the same because sentences that are translations of each other have similar lengths. This is measured in terms of the length of the sentences, ratio of the lengths, and difference in the lengths.

2. **Word Overlap**

   We find out the percentage of words in each sentence that have a translation in the corresponding sentence on the other side. This translation is considered according to the dictionary (lexicon), and words are weighted by their inverse frequency in the document. So, less frequent words add more value to the score.

3. **Relative position in document**

   We consider the difference in the relative position of the sentences in their respective documents. This is based on the observation that aligned articles have a similar topic progression. So, if a sentence is in the middle of the document in source language, its translation can be found roughly in the same region in the corresponding target language document.

4.5.2 Word Alignment based Features

Many aspects of word alignments computed between a pair of sentences can be useful in deciding whether they are translations of each other. These can be used as features for our classification process.

1. **Number of aligned words**

   The percentage of the number of words that are aligned in the sentence pair is a good indicator of parallelism. It a large percentage of words is unaligned, then the sentences are not likely to be parallel.

2. **Length of contiguous connected spans**

   A contiguous connected span is a long sequence of words in the sentence which are all aligned to a sequence of words in the corresponding sentence on the other side. This feature is also indicative of parallelism. Longer the length of such a contiguous connected span, greater are the chances that the two sentences are parallel to each other.

3. **Top three largest fertilities**

In the alignment computed between a sentence pair, it a single word in one sentence has an alignment of large fertility, it can indicate non-parallelism. This is because, in sentences which are translations of each other, words usually have low fertility and the number of words with high fertility is also less. So, we can find the top three largest fertilities in the alignment between the sentence pair to decide whether they are parallel.

4. **Length of contiguous unconnected spans**

   An unconnected span is similar to the above-mentioned connected span. The only difference is that it has a long sequence of unaligned (unconnected) words. If the length of the longest unconnected span in a sentence is very large (more than say, 1/3rd of the sentence length), then it indicates non-parallelism.

4.5.3 Distortion Features

As we use distortion model in a word or a phrase-based translation model for a sentence, we can also consider a distortion of sentences in a document. That is, we consider the difference between the positions of the previous and the current pair of aligned sentences.

4.6 LEXACC (Lucene Based Parallel Sentence Extraction from Comparable Corpora)

The LEXACC system (Stefănescu et al., 2012), uses Cross Language Information Retrieval in order to reduce the search space for finding parallel sentences in comparable corpora. First, target sentences are considered as documents, and are indexed.Then, for each source sentence, the content words are translated into the target language according to a dictionary. The translations are used to form a Boolean query which is then fed to the search engine and the top hits are considered to be translation candidates.

These candidates are then compared to the source sentence using a similarity measure (described later), to finally say whether or not they are parallel.

4.6.1 Indexing, Searching and Filtering

The target sentences are first transformed such that only stemmed non-functional words are present in them. Then, the transformed target sentences are indexed as documents using Lucene. Additional searchable fields are also employed, namely, *short* (if sentence length is short), *long* (if sentence is long), and an optional term *document* (the document to which the sentence belongs in the comparable corpus).

The searching step involves the use of a GIZA++ constructed dictionary for constructing the query. Each non-functional word is translated using the dictionary and added to the query as a disjunctive query term (SHOULD occur). Two other disjunctive terms added to the query are *short* and *long*, according to what is applicable to the source sentence. If the optional term *document* is known, then it is added as a conjunctive term (MUST occur) to the query.

Filtering of each candidate pair is done to further reduce the search space. A *viability score* is computed and a threshold is used to filter the candidate sentence pairs.

$$viabilityScore = \alpha \cdot \beta \cdot se \cdot sim$$

$$\alpha = 1 - \frac{abs(|s| - |t|)}{max(|s|, |t|)} \tag{4.3}$$

$$\beta = \frac{min(|s|, |t|)}{\lambda} \tag{4.4}$$

where *se* is a score returned by the search engine. —*s*— and —*t*— are lengths of s and t. λ is the threshold for considering sentences as long.

$$sim = \frac{2 \cdot teFound \cdot te}{|s| + |t|} \cdot \frac{1}{\sqrt{coh}} \tag{4.5}$$

where *teFound* is the total number of words in s for which we found translation equivalents in t, *coh* is the cohesion score computed as the average distance between the sorted positions of the translation equivalents found in t (the lower the better)5 and te is calculated as:

$$te(s,t) = \sum_{w_s \epsilon t} \max_{w_t \epsilon t} dicScore(w_s, w_t) \tag{4.6}$$

where *dicScore* is the translation probability from dictionary.

4.6.2 Translation Similarity Measure

The translation similarity measure is a weighted sum of feature functions that indicate if s is translated by t :

$$P(s,t) = \sum_i \theta_i f_i(s,t) \text{ ; such that } \sum_i \theta_i = 1$$

Features

1. **Content words translation strength:** Find the best 1:1 alignment using a Giza++ lexicon, and find the ratio of addition of translation probabilities of the translated words and length of sentence. This is specifically for content words only.

2. **Functional words translation strength:** Finding the same score, but for function words.

3. **Alignment Obliqueness:** It is a discounted correlation measure because languages naturally have different word orderings, hence, crossings of alignment links.

4. **Strong Translation Sentinels:** If sentences are parallel, there are content word translations near the beginning and end of the sentences.

5. **End with same punctuation:** Parallel sentences end with the same type of punctuation.

Learning Optimal Weights

A standard logistic regression classifier was trained to obtain a set of optimal weights on positive and negative examples.

Summary

In this chapter, we looked at methods based on classification, ranking, similarity, discriminative modeling, etc. for finding parallel sentences from comparable documents. Most of these techniques make use of word alignment based features or similarity scores to distinguish parallel sentences from non-parallel. In the next chapter, we will discuss methods of parallel phrases or parallel fragments extraction.

Chapter 5

Extraction of Parallel Sub-Sentential Fragments

When the degree of parallelism of the comparable corpus is quite low, looking for absolutely parallel sentences may not yield good results. This is because, such parallel sentence pairs may not be present in the corpus, at all. But, if we go down to a finer granularity of phrases or segments of sentences, there is a higher possibility of finding good translations even in very non-parallel corpora, than at the sentence level.

Identifying such sub-sentential parallel fragments is a difficult task since it involves recognizing an overlap in sentence pairs that may express very different content. This section presents the existing work on finding parallel segments in comparable corpora.

5.1 Log-Likelihood Ratio

The approach based on Log-Likelihood Ratio is described by Munteanu and Marcu (2006). The input to this technique is a set of candidate sentence pairs that can be obtained by filtering those sentence pairs that have a high word overlap from all possible sentence pairs in the corpus. This is done using a lexicon obtained by running GIZA++ on an initial parallel corpus. This step does not need to be very precise, but needs a higher recall.

The next step focuses on finding fragments of the source sentence that have a translation on the target side. In this step, the lexicon obtained by GIZA++ is not very useful because such a lexicon contains entries for even unrelated word pairs. These incorrect correspondences can adversely affect the results that we obtain from this step. So, precision is of utmost importance here. Hence, we also need a measure of two words not being translations of each other. The Log-Likelihood Ratio (LLR) statistic is used for obtaining such a cleaner lexicon that does not contain any incorrect correspondences.

5.1.1 LLR to Estimate Word Translation Probabilities:

LLR gives the measure of the likelihood that two samples are not independent. This measure is used to find the independence of all pairs of co-ocurring words in the initial parallel corpus. If source word f and target word e are independent, then $\mathrm{p}(e|f) = \mathrm{p}(e|{\sim}f) = \mathrm{p}(e)$. So, if the words are independent, i.e., these distributions are very similar, the LLR score of this word pair is low. If the words are strongly associated, then the LLR score is high.

But, a high LLR score implies either positive correspondence ($\mathrm{p}(e|f) > \mathrm{p}(e|{\sim}f)$) or a negative correspondence ($\mathrm{p}(e|f) < \mathrm{p}(e|{\sim}f)$) between the words. So, the set of co-occurring word pairs in the parallel corpus, is split into two sets: positively associated and negatively associated word pairs. Note that, here, co-occurring words are those that are linked together in the word-aligned parallel corpus. Thus, the LLR score is enhanced by this notion of co-occurrence using the knowledge of word alignment in a parallel corpus.

LLR(e,f) is computed for each of the linked word pairs and then, two conditional probability distributions are computed:

- $P^{+}(e|f)$ is probability that source word f gets translated to target word e
- $P^{-}(e|f)$ is probability that source word f does not get translated to target word e

Similarly, the probability distribution in the reverse direction is also computed.

5.1.2 Detecting Parallel Fragments:

The target sentence is considered as a numeric signal. The translated words give positive signals (from P^+ distribution) and untranslated words give negative signals (from P^-) distribution. So, only that part which is positive, is retained as the parallel fragment of the sentence. For each linked target word, the value of the signal is the probability of its alignment link $P^+(e|f)$. All the remaining unaligned target words have signal value $P^-(e|f)$. This forms the *initial signal*.

Then, a *filtering signal* is obtained by averaging the signal values of nearby points. The number of points to be used for averaging is decided empirically. Then, the "positive signal fragment" of the sentence is retained. This approach tends to produce very short fragments. So, fragments less than 3 words in length can be discarded. The procedure can be repeated in the opposite direction and the results can be symmetrized.

5.2 Sentence Splitting for Phrase Alignment

Vogel (2005) describes an algorithm PESA for phrase alignment in parallel sentences by posing it as a problem of sentence splitting, and has derived a maximum likelihood estimation expression for it. Hewavitharana and Vogel (2011) used the same technique to extract parallel phrases from comparable corpora. This section describes this technique in detail and then we describe our experiments and evaluation of this technique.

The standard method of finding phrase translations in a parallel corpus is to find the word alignments in both directions (source to target and target to source) and then combine them using various heuristics. This kind of phrase alignment is essentially a post-processing step to word alignment. In this method of sentence splitting, we do not search for all possible phrase alignments in a sentence pair. Instead, a simpler problem is formulated.

Suppose $\mathbf{f} = f_1, f_2, ...f_l$ is a source phrase. We should find a sequence of words i.e., a target phrase $\mathbf{e} = e_1, e_2, ...e_k$ in the target sentence such that the phrase pair $\mathbf{e}$ and $\mathbf{f}$ are translations of each other.

5.2.1 Constrained IBM Model 1

The PESA algorithm identifies the boundaries of the target phrase for a given source phrase. The words inside the source phrase are aligned to words inside the target phrase, and the words outside the source phrase are aligned to words outside the target phrase.

The position alignment probability for a sentence is $1/I$, where $I =$ length of target sentence and J is the length of the source sentence. This is changed to be $1/k$ inside the source phrase and $1/(I-k)$ outside the source phrase, where $k =$ length of target phrase.

The alignment probability is given by:

$$p(s \mid t) = (\prod_{j=1}^{j_1-1} \sum_{i \notin (i_1 ... i_2)} \frac{1}{I-k} p(s_j \mid t_i)) \times (\prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} \frac{1}{k} p(s_j \mid t_i))$$

$$\times(\prod_{j=j_2+1}^{J} \sum_{i \notin (i_1 ... i_2)} \frac{1}{I-k} p(s_j \mid t_i)) \tag{5.1}$$

For symmetrization, $p(t \mid s)$ is similarly calculated in the reverse direction as:

$$p(t \mid s) = (\prod_{i=1}^{i_1-1} \sum_{j \notin (j_1 ... j_2)} \frac{1}{J-l} p(t_i \mid s_j)) \times (\prod_{i=i_1}^{i_2} \sum_{j=j_1)^{j_2}} \frac{1}{l} p(t_i \mid s_j))$$

$$\times(\prod_{i=i_2+1}^{I} \sum_{j \notin (j_1 ... j_2)} \frac{1}{J-l} p(t_i \mid s_j)) \tag{5.2}$$

where l is the length of the source phrase.

The optimal target boundaries are found by interpolating the probabilities given by equations 5.1 and 5.2. The MLE expression selects the boundary (i_1, i_2) which maximizes the interpolated probability.

$$(i_1, i_2) = \arg\max_{i_1, i_2} \; (1 - \lambda) \cdot log(p(s \mid t)) + \lambda \cdot log(p(t \mid s)) \tag{5.3}$$

The value of λ can be tuned from a set of held-out data. This algorithm uses a seed lexicon for finding the translation probabilities in the computation.

5.3 Chunking Based Approach

Extracting parallel fragments of text using *chunking*, is one of the most recent advances that have been made in this area by Gupta et al. (2013). In this approach, a document-aligned comparable corpus is used, and all possible sentence pairs within each aligned document pair are examined to look for parallel segments of sentences. The sentences were broken into fragments and then, we check which of the fragments have a translation on the target side.

But, instead of segmenting the source sentence into N-grams, *chunking* is used to obtain *linguistic phrases* from the source sentences. According to linguistic theory, the tokens within a chunk do not contribute towards long distance reordering, when translated. That means, the entire chunk's position can be changed within the translated sentence, but the words within a chunk remain within the chunk. But, ad-hoc N-gram segments may not be linguistic phrases, and are always of constant length. Chunks are are variable length and chunks can be merged to form larger chunks or even sentences.

5.3.1 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.

5.3.2 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:

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

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.

5.3.3 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.

Summary

This chapter introduced techniques based on Log-Likelihood Ratio, and chunking for parallel phrase extraction from comparable corpora. The next chapter discusses techniques for bilingual lexicon extraction from comparable corpora.

Chapter 6

Bilingual Lexicon Extraction

Extracting bilingual lexicons is one of the oldest ways of exploiting comparable corpora. There have been many different approaches for extracting lexicons, named entities, terminologies, etc. from comparable and non-parallel corpora. Some of them have been discussed in this chapter.

6.1 Bootstrapping

The technique of extracting a bilingual lexicon from comparable to very non parallel corpora is described by Fung and Cheung (2004a). They first extract parallel sentences from similar documents using the cosine similarity measure. Then, the IBM Model 4 EM learning is applied on these extracted set of parallel sentences to find unknown word translations.

GIZA++ is used for this purpose. But this method is used on the results of parallel sentence extraction. So, the set of aligned sentence pairs may also contain many pairs that might not really be parallel. The alignment scores that are computed by GIZA++ on extracted parallel sentences do not necessarily correspond to the actual similarity of the sentence pair, because, EM estimation is weak when applied to a small set of bilingual sentences. One solution to this problem is to initialize the EM estimation using a corpus of truly parallel data.

Thus, this step is used to enhance the existing lexicon, which in turn, is used to extract more parallel sentences. Bootstrapping is done in this way until no new sentence pairs are obtained.

6.2 EM based Hybrid Model

Bilingual terminology extraction using an unsupervised hybrid model incorporating many features in an EM-based framework is described by Lee et al. (2010). The motivation for this work is that most of the dictionaries and lexicons contain many words, but very few terminologies. The given set of comparable corpora is noisy and has a variety of domains and topics, so first, document alignment is performed. Documents are associated based on common words and numbers and identical strings in the title, content and also, their distribution in time.

After obtaining aligned pairs of documents, they are tokenized to find out the prominent noun terms. Thus, we have a set of aligned document pairs and a set of noun terms for each document in the corpus. In comparable corpora, there is a lot of noise and also, significant differences in the structure of sentences. So, features are obtained from corpus-driven as well as non-corpus-driven information, and these features are used to initialize the score of each term pair candidate.

6.2.1 The EM Algorithm

The state-of-the-art IBM Model 1 EM algorithm is used to compute alignment between terms of a document pair. In the reformulation of the algorithm, comparable document pairs take the place of parallel sentences, and are initialized with the document alignment score. So, this non-uniform initialization compensates for the corpus not being parallel. Also, the feature set explained below, is also used for initialization of term alignment.

EM of IBM Model 1 works well for parallel sentences, but not for comparable documents, because it converges to a global maximum and comparable data is too noisy.

So, to avoid overfitting, only 10 iterations were run in both directions (f to e and e to f). After each iteration, weak alignments are filtered out using a high threshold of 0.8.

6.2.2 Similarity Features

Following feature function scores are used for candidate alignment initialization:

1. Document Alignment Score (D): This score indicates the confidence level of document alignment, normalized to the range of [0,1].

2. Lexical Similarity (L): If a candidate pair of terms occurs with each other more than 50% of the times, than any other word, then this pair is given a score of L_O, and 1 otherwise. $L_O > 1$ is a configurable weight that can be given to lexical similarity.

3. Named Entity Similarity (N): We can have a predefined set of categories to which a named entity can belong to, like, person, place, organization, etc. Then, **N** is the likelihood of that the two terms in a candidate pair belong to the same named entity category. If they match, then the score is N_O >1 and 1 otherwise.

4. Context Similarity (C): Terms with similar context are likely to have similar meanings. So, we find the context similarity of the term pair using *'k'* nearest content words within the sentence boundary and finding the cosine similarity between their context frequency vectors.

5. Temporal Similarity (T): This is a useful feature if the comparable corpus is from newspaper domain and alike. Terms occurring in such corpora are usually synchronous in time, if they truly correspond to each other. '**T**' is calculated by first finding its frequency *spectrum* in discrete time. Then, the *power spectrum* (the magnitude square of the *spectrum*) of two terms is used to find temporal similar-

ity between them, by means of cosine similarity. *Power spectrum* is sensitive to relative spacing in time but not to shifting in time.

6. Related Term Similarity (R): Related terms correlate statistically in the same documents. They can be found using t-test or mutual information in a monolingual corpus. This is a measure of related term likelihood. Truly aligned terms have similar related terms.

The EM algorithm, modified to use all the above mentioned feature functions, converges to give an aligned set of terms from the comparable corpus. It is shown to work well on comparable corpora, but this framework works better on a parallel corpus. The performance of this technique is dependent on the degree of parallelism of the comparable corpus to a large extent.

6.3 Pivot Language and Word Alignment

There are many languages like Hindi, French, Chinese, etc. which are very less known as a language pair, but have considerable usage with a common language like English. Kwon et al. (2013) describe a technique for extracting a bilingual lexicon for such a rare language pair by using a *pivot language* combined with information retrieval techniques and word alignment.

6.3.1 Approach

Instead of comparable corpora, two parallel corpora are used. Let L_1 and L_2 be the two languages for which we desire a bilingual lexicon and L_P be the pivot language. So, we use a $L_1 - L_P$ parallel corpus and a $L_P - L_2$ parallel corpus. For example, let L_1 be Korean, L_2 be Spanish, and L_P be English. So, we need a Korean-English parallel corpus and an English-Spanish parallel corpus. Following steps are performed for building the Korean-Spanish lexicon:

1. Align words in the Korean-English corpus and the English-Spanish corpus. All words in Korean and Spanish now have a vector of corresponding English words with their associated alignment scores.

2. For each Korean word, similarity of its English word vector with the English word vectors of each Spanish words is computed by means of cosine similarity, dice coefficient or Jacquard's Similarity.

3. A list of top *'k'* word pairs is obtained as the lexicon for Korean-Spanish.

This technique does not use any resources of the L_1 and L_2 language pair. Instead, it needs a parallel corpus of these languages with a pivot language like English.

6.4 Canonical Correlation Analysis and EM

An EM based approach that employs Canonical Correlation Analysis is described by Haghighi et al. (2008) for bilingual lexicon extraction from monolingual corpora.

6.4.1 Generative model for Bilingual Lexicon Induction

The input to this approach is are source and target monolingual corpora S and T. Let $\mathbf{s}$ = $(s_1, ..., s_{n_s})$ words in S and $\mathbf{t} = (t_1, ..., t_{n_t})$ words in T. The goal is to find a matching $\mathbf{m}$ between $\mathbf{s}$ and $\mathbf{t}$. The matching $\mathbf{m}$ is represented as a set of integer pairs such that *(i, j)* ϵ $\mathbf{m}$ if s_i is matched with t_j. For each matched pair of words *(i, j)* ϵ $\mathbf{m}$, observed feature vectors of source and target words: $f_S(s_i)$ and $f_T(t_j)$, are generated from the corresponding monolingual corpus. These features summarize the word's monolingual characteristics.

If *(i, j)* ϵ $\mathbf{m}$, then s_i and t_j are considered to be translations of each other. So, we can expect that their feature vectors will be similar and somehow connected by the generative process. Let them be related through the vector $z_{i,j}$ that represents the shared, language independent concept.

To generate the feature vectors, first, a random concept $z_{i,j} \sim \mathcal{N}(0, I_d)$, where I_d is the $d \times d$ identity matrix. Then, the source feature vector $f_S(s_i)$ is drawn from a multivariate Gaussian with mean $W_S z_{i,j}$ and covariance Ψ_S. W_S is a $d_S \times d$ matrix which transforms the language independent concept $Z_{i,j}$ into a language dependent vector $f_S(s_i)$ in the source space. The target $f_T(t_j)$ is generated analogously and conditionally independent of the source given zi, j, using W_T and Ψ_T.

6.4.2 EM Framework of Inference

The objective is to maximize the log-likelihood of the observed data (**s**,**t**):

$$l(\theta) = \log p(\mathbf{s}, \mathbf{t}; \theta) = \log \sum_m p(\mathbf{m}, \mathbf{s}, \mathbf{t}; \theta)$$

with respect to the model parameters $\theta = (W_S, W_T, \Psi_S, \Psi_T)$.

In the EM algorithm, we find a maximum weighted partial bipartite matching **m** in the E-Step. In the M-Step, we find the parameters θ by performing canonical correlation analysis(CCA).

- **E-Step:**

  $$\mathbf{m} = \arg\max_{m'} \log p(m', s, t; \theta)$$

  This is cast as a maximum weighted bipartite matching problem and can be solved using the Hungarian Algorithm.

- **M-Step:** Given a matching **m**, the M-Step optimized log p(**m**,**s**, **t**; θ) with respect to θ.

  $$\max_\theta \sum_{(i,j)\epsilon\mathbf{m}} \log p(s_i, t_j; \theta)$$

  This objective is similar to maximizing the likelihood of a probabilistic CCA model and it is proved that maximum likelihood estimate can be computed using CCA.

So, CCA finds d-dimensional subspaces of source and target so that components of projection are maximally correlated.

6.4.3 Feature Set:

Following features are used to generate the feature vector $f_S(.)$ and $f_T(.)$, which are derived from the monolingual corpora.

1. **Orthographic features:** This is more suitable for closely related languages, as they share many orthographic features. *Edit Distance* is used to get a score for this feature.

2. **Context Features:** If words s and t are translations of each other, then there also should be a strong correlation between the source words in the context of s and target words in the context of t.

Thus, a bilingual lexicon is generated by first having a language independent concept represented in the latent space and then generating the feature vectors that connect the words in different languages to this concept. These words are nothing but translations of each other.

Summary

This chapter explored various approaches for extracting bilingual lexicons from comparable corpora. Bilingual Lexicons can be learned using approaches like bootstrapping, a feature-rich hybrid EM framework, making use of a Pivot language, canonical correlation analysis, etc. The next chapter explains bootstrapping to make the most out of a comparable corpus.

Chapter 7

Bootstrapping

The number of parallel sentences that can be extracted from comparable or non-parallel corpora depends on the bilingual lexicon (dictionary) or the phrase translation table generated by training the classifiers on a seed parallel corpus. If this seed corpus has a poor dictionary coverage, then the parallel sentence extraction process can be adversely affected. This is because, many words from the candidate sentence pairs will be out-of-vocabulary for our translation table, and hence, the sentence pair may be classified as non-parallel even if it is very parallel.

A simple solution to this problem is bootstrapping (Fung and Cheung, 2004a). In bootstrapping, we perform parallel sentence extraction in iterations. In each iteration, we learn new word translations from the intermediate output of the previous parallel sentence extraction. This refines our bilingual lexicon with more in-domain data, and hence the overall extraction process.

Steps for bootstrapping:

- Document Alignment

  Find all the similar document pairs from the corpus using any of the techniques discussed previously. These mostly employ word overlap, TFIDF and cosine similarity.

- Parallel Sentence Extraction

  In the aligned documents, find all parallel sentences using any of the techniques

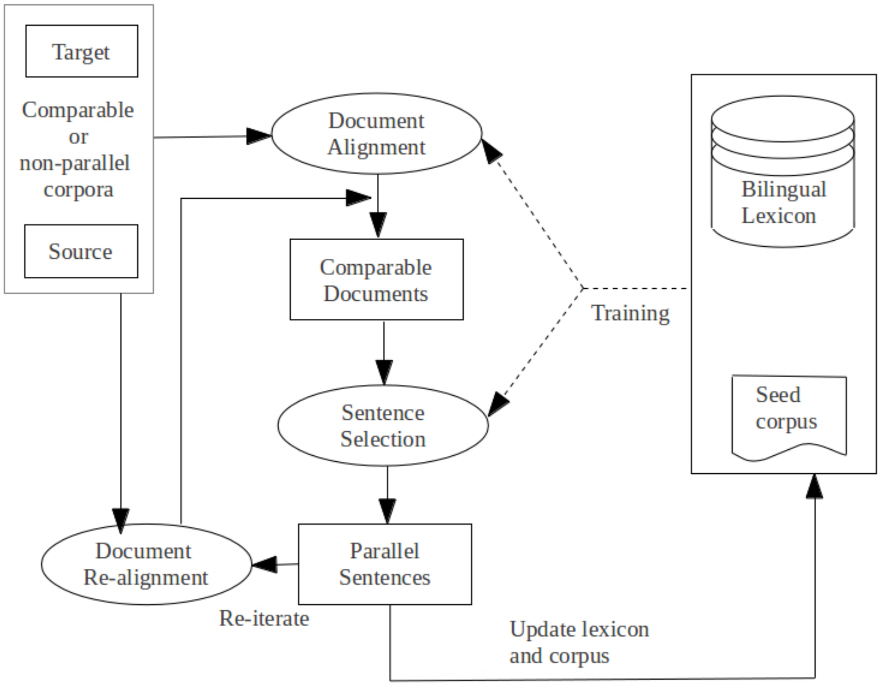


Figure 7.1: Parallel Sentence Extraction System with Bootstrapping

discussed in the previous sections. They rely mostly on word overlap, ME classifiers or Ranking models and sentence similarity.

- Update Lexicon

  The extracted sentence pairs can be simply added to the training data and an updated bilingual lexicon can be obtained.

- Iterate

  After obtaining the updated lexicon, we reiterate from the document alignment step.

- Terminate

  The iteration process can be terminated when no new parallel sentences are extracted.

7.1 Issues with Bootstrapping

A bootstrapping system is able to learn new lexical items that occur only in the extracted parallel sentences. But, the number of sentences extracted may be quite low as compared to the entire collection of documents from which we are extracting. This collection of documents may also contain many aligned document pairs that do not contain any parallel sentences, yet contain many words and phrases that are good translations of each other.

7.2 Alternative Approach: Lexicon Induction

To take advantage of such word and phrase occurrences, we can use an alternative approach (Smith et al., 2010) of lexicon refinement. A Lexicon model based on

$P(w_t|w_s,T,S)\Psi$

where, w_t *is word in target language*

w_s *is word in source language*

T *and* S *are target and source languages.*

This model is trained similar to the sentence extraction model, but the only difference is that, we will align word pairs instead of sentence pairs with this model. It can be trained using a corpus that has annotated alignments for the word that are translations of each other. Then, the following *word-level induced lexicon* features (Smith et al., 2010) can be used in the lexicon model.

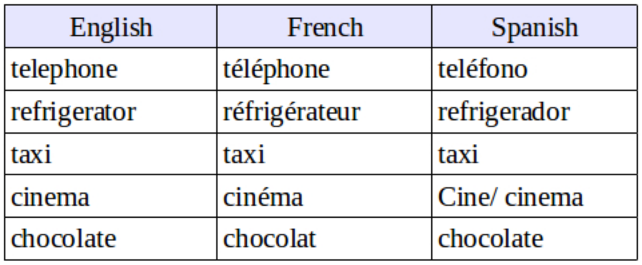
7.2.1 Translation probability

An HMM word alignment model can be trained on some seed parallel data and the translation probabilities can be taken as features. The translation probabilities are p($w_t|w_s$), the opposite direction *i.e.*, p($w_s|w_t$) and also log of these probabilities.

7.2.2 Orthographic Similarity

Orthographic similarity can be calculated using a function of the edit distance (Levenshtein distance) between the source and target words. Edit distance is the minimum number of single-character edits required to change one word into the other. A small edit distance generally implies that the words are translations of each other.

e.g. consider the set of English and corresponding French and Spanish words shown in figure 7.2. Each of them are translations, and have a small edit distance (2 to 3 letters).



English	French	Spanish
telephone	téléphone	teléfono
refrigerator	réfrigérateur	refrigerador
taxi	taxi	taxi
cinema	cinéma	Cine/ cinema
chocolate	chocolat	chocolate

Figure 7.2: Levenshtein distance

7.2.3 Context Similarity

This feature measures the context similarity of two words w_s and w_t in source and target languages, respectively. All the words occurring next to w_s in article S and all the words occurring next to w_t in article T, within a context window, are found out. Then, scoring functions are computed to measure the translation correspondence between these contexts. A similar context can be indicative of the words w_s and w_t being translations of each other.

7.2.4 Distributional Similarity

This feature corresponds closely to the *context similarity* feature described above. For each source word w_s, we collect a distribution of context words v_s in offset positions, $o \, \epsilon$ {-2,-1,+1,+2}. -2 and -1 are two and one positions to the left of w_s respectively, and similarly, +1 and +2 are one and two positions to the right. This distribution is based on the count of the number of times a context word v_s occurs in a position o in the context of w_s. The probability distribution is given as:

$$\begin{aligned} &P(o,v_s \mid w_s) \; \alpha \; weight(o) \cdot count(w_s,o,v_s) \\ &weight(o) = 2 \cdots \; if \; o = \pm 1 \\ &\qquad\qquad\;\; = 1 \cdots \; otherwise \end{aligned}$$

A similar distribution of context words v_t is collected for all target words w_t. Then, we use an IBM Model 1 trained on a seed parallel corpus and get a translation table P($v_t \mid v_s$) *i.e.*, the translation probability of context words. Using this, we estimate a cross-lingual context distribution as,

$$P(o,v_t \mid w_s) = \textstyle\sum_{v_s} P(v_t \mid v_s).P(o,v_s \mid w_s).$$

Then, the similarity of words w_s and w_t is found by using

Similarity of w_s and w_t = 1 Jensen-Shannon divergence of the distributions over positions and target words.

Thus, either bootstrapping or a word level lexicon model can be used for acquiring new lexical items at each intermediate step. This helps in more precise re-alignment of documents and extraction of parallel sentences.

Summary

Bootstrapping can be employed to increase the number of sentences that can be extracted. But, bootstrapping has some issues, which can be solved using an induced lexicon.

Chapter 8

Conclusion

Lack of parallel corpora is a major bottleneck in development of SMT systems for many language pairs. There are large amounts of comparable and non-parallel corpora available for many language pairs. These can be exploited to extract parallel data from them.

From any non-parallel set of documents, document alignment gives a definite region in the entire search space to look for parallel sentence pairs. The sentence extraction step must be robust in order to filter any noise and give accurate, parallel sentence pairs. Good use of feature functions for ME classifiers and ranking models can be used to serve the purpose.

The amount of data extracted is often adversely affected by a poor dictionary coverage. This can easily be resolved by using a few boostrapping iterations or other approaches as discussed.

Data extracted by such techniques tends to be somewhat noisy but is still shown to be very useful in a standard SMT system. Thus, this is a great step in creation of parallel corpus for language pairs that have scarcity of parallel corpora resources.

Bibliography

Blunsom, P. and Cohn, T. (2006). Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72. Association for Computational Linguistics.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

Fung, P. and Cheung, P. (2004a). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of EMNLP*, volume 2004.

Fung, P. and Cheung, P. (2004b). Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1051. Association for Computational Linguistics.

Gupta, R., Pal, S., and Bandyopadhyay, S. (2013). Improving mt system using extracted parallel fragments of text from comparable corpora. In *Proceedings of the 6th*

Workshop on Building and Using Comparable Corpora, pages 69–76. Association for Computational Linguistics.

Haghighi, A., Liang, P., Berg-Kirkpatrick, T., and Klein, D. (2008). Learning bilingual lexicons from monolingual corpora. In *ACL*, volume 2008, pages 771–779.

Hewavitharana, S. and Vogel, S. (2011). Extracting parallel phrases from comparable data. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 61–68. Association for Computational Linguistics.

Koehn, P. (2010). *Statistical Machine Translation.* Cambridge University Press.

Kwon, H.-S., Seo, H.-W., and Kim, J.-H. (2013). Bilingual lexicon extraction via pivot language and word alignment tool. In *Proceedings of the 6th Workshop on Building and Using Comparable Corpora*, pages 11–15. Association for Computational Linguistics.

Lee, L., Aw, A., Zhang, M., and Li, H. (2010). Em-based hybrid model for bilingual terminology extraction from comparable corpora. In *Poster Volume*, pages 639–646. Coling.

Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*, volume 999. MIT Press.

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*, pages 81–88. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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

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*, pages 403–411. Association for Computational Linguistics.

Stefănescu, D., Ion, R., and Hunsicker, S. (2012). Hybrid parallel sentence mining from comparable corpora. In *Proceedings of the 16th Conference of the European Association for Machine Translation*, pages 137–144.

Vogel, S. (2005). Pesa: Phrase pair extraction as sentence splitting. In *Proc. of the Machine Translation Summit*, pages 251–258.

Vogel, S., Ney, H., and Tillmann, C. (1996). Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.