ON-LINE COMPILATION OF COMPARABLE CORPORA AND THEIR EVALUATION

Radu ION, Dan TUFIŞ, Tiberiu BOROŞ, Alexandru CEAUŞU, and Dan ŞTEFĂNESCU

Research Institute for Artificial Intelligence, Romanian Academy

ABSTRACT

Using comparable corpora is became a topic in the mainstream Machine Translation (MT) research because, for less resourced languages, mining the Web for comparable corpora is assumed to be more productive than searching for parallel corpora. The experiments in using comparable corpora in enhancing translation models demonstrated significant improvements in MT accuracy. This paper reports on specific procedures of building comparable corpora from Wikipedia and from general Web using a highly customizable application that can merge diverse web crawlers and source their output either into files or NLP web services. We also describe a method of scoring a pair of documents from a comparable corpus as to their parallelism degree.

Introduction

Multilingual comparable corpora (MCC) have been around for a while in the context of Machine Translation (MT) research, as an alternative to parallel corpora which were (and still are for certain pairs of languages and domains) hard to find. By comparison with parallel corpora which contain pairs of equivalent translation units of text (sentences or paragraphs), MCC exist with different degrees of comparability: weakly comparable corpora, strongly comparable corpora, quasi-comparable corpora, very-non-comparable corpora, etc. (Skadiņa et al. 2010). A general definition of MCC that we find operational is given by (Munteanu and Marcu, 2006). They say that a (bilingual) comparable corpus is a set of paired documents that, while not parallel in the strict sense, *are related* and *convey overlapping information*. The measure of this overlapping should give the degree of the comparability between the two documents in a pair (for instance, a real number ranging between 0 and 1 with 0 indicating complete divergence of topic and 1 indicating parallelism: one document is the translation of the other).

Systematic research on methods for building and exploiting MCC is relatively new and several relevant papers can be found in the proceedings of the 1st, 2nd and 3rd workshops on Building and Using MCC: http://www.limsi.fr/~pz/lrec2008-comparable-corpora/ (LREC2008), http://comparable2009.ust.hk/ (ACL-IJCNLP 2009) and http://www.fb06.uni-mainz.de/lk/bucc2010/(LREC 2010).

There are methods of sentence alignment, named entity and terminology translation, extracting bilingual dictionaries, studying the effect of using comparable corpora on the accuracy of MT, all using comparable corpora. While the accent is naturally on the particular algorithm or model described, little or nothing special is said about the compilation of the MCC that was used. Available algorithms of collecting MCC always refer to methods to *ascertain the degree of comparability* that exists between two topically related documents. Typically, one starts with a collection of terms from a given domain along with their translation in the target language, retrieves two sets of documents corresponding to the source and the target terms and then decides which pair(s) of documents should be added to the MCC of that domain. If the sets are large (1000 documents for instance), one should have at his/her disposal a fast algorithm that will process all pairs of documents (in our example 1,000,000 documents). Therefore, having a website such as English Wikipedia in which every article is categorized and is also linked with its foreign version, be it merely a translation or otherwise a complete rethinking of the subject, constitutes an immense advantage. Thus, Wikipedia is an already established and very good quality comparable corpus[1] of many domains and the task of constructing the collection of documents pairs is greatly simplified by its structure.

In what follows, we present our MCC harvesting algorithms and applications, reporting on the sizes of corpora that have been obtained. We will sketch a scoring algorithm for computing the comparability degree of an arbitrary pair of documents, a function that is most useful in building MCC when the pairing of the documents is unknown.

Collecting Comparable Corpora from Wikipedia

For building a strongly comparable corpus one step is to identify pairs of documents that are topically related. Work in this direction is reported in Munteanu (2006) who describes a method of identifying parallel fragments in MCC. Another

[1] At least, if we speak of quality articles of this website.

experiment in pairing topically related documents is due to Tao and Zhai (2005).They tackle the problem of MCC acquisition by devising a language independent method based on the frequency correlation of words occurring in documents belonging to a given time scale. The intuition is that two words in languages A and B whose relative frequency vectors Pearson-correlate over n pairs of documents in languages A and B that are paired by a time point i, are translation equivalents. This relative frequency correlation is then used as a translation equivalence association score of words in languages A and B for describing a measure of document relatedness. Vu et al. (2009) improve the accuracy of method described above by a margin of 4% on an English-Chinese corpus. Wikipedia as a comparable corpus has been studied and used by Yu and Tsujii (2009). They sketch a simple mining algorithm for MCC, exploiting the existence of inter-lingual links between articles.

Our goal is to extract good quality MCC in languages Romanian, English and German for use in the ACCURAT project[2]. We have employed two different methods of gathering MCC from Wikipedia:

1. the first one considers an input list of good quality Romanian articles (articles that senior Wikipedia moderators and the Romanian Wikipedia community think that they are complete, well written, with good references, etc.) from the Romanian Wikipedia (http://ro.wikipedia.org/) and for each such article, it searches for the equivalent in the English Wikipedia;
2. the second one uses the Princeton WordNet and extracts all the capitalized nouns (single-word or multi-word expressions) from all the synsets. Then, it looks for Wikipedia page names formed with these nouns, extracts them and their correspondent Wikipedia pages in Romanian and German (it these exist).

The first method of MCC compilation uses 3 different heuristics of identifying the English equivalent of a given Romanian article (they are tried in the listed order):

a) it searches for an English page with the exact name as the Romanian page. For instance, we have found the following exact-match English pages (starting from the Romanian equivalents): “Alicia Keys”, “Hollaback Girl”, etc.;

b) it searches for the English link from the Romanian page that would lead to the same article in those languages. The Romanian version of the page may or may not be a complete translation from English (we noticed that the translation is usually shuffled – the narrative order of the English page is rarely kept and it usually reflects the translator’s beliefs with regard to the content of the English page);

c) it automatically transforms the Romanian page name into an English Wikipedia search query by using a translation dictionary that has scores for each translation pair. Thus, for each content word in the Romanian page name, generates the first *k* translations (*k*=2 in our experiments) and with this query, retrieves the first 10 documents from the English Wikipedia. We manually chose the right English candidate but an automatic pairing method based on document clustering is described below.

Using these heuristics, we managed to compile a very good Romanian-English comparable corpus that consists of 128 paired Romanian and English documents of approx. 502K words in English and 602K words in Romanian.

The second method of MCC compilation uses Princeton Wordnet for extracting a list of named entities. These named entities are then transformed into Wikipedia links by replacing the white spaces with underscore and adding the string “http://en.wikipedia.org/wiki/” in front of them. Then, an application performs the following steps:

a) it goes to every link and downloads the Wikipedia page if it exits;
b) every downloaded Wiki page is searched for links to correspondent Romanian and German Wiki pages; if such links exist, those pages are also downloaded;
c) all the html tags of every En-Ro or En-De pair of Wiki documents are stripped so that only the plain text remains (there is also the possibility of preserving some mark-ups for important terms highlighted in Wikipedia articles); The categories of the documents are kept in a simple database.

Using the categories of the documents one can select documents referring to specific subjects. However, due of the fact that we searched only for named entities, confusions might occur. For example, Wiki articles about Paris, Rome or London might be considered to be about sports as they are categorized, among others, as “Host cities of the Summer Olympic Games”. In reality, these articles contain very few information about such a topic. The Table 1 shows the amount of comparable data we

[2] http://www.accurat-project.eu/

extracted from Wikipedia using the described method.

Named Entities pages about:	en-ro	de-ro
Sports	1043.9 K	534.1 K
Software	63.3 K	35.8 K
Medical	617.7 K	400.9 K
Other	43,965 K	25,042.8 K
Total	**45,689.9 K (418.3 Mb)**	**26,013.6 K (239.2 Mb)**

Table 1: The amount of comparable data extracted from Wikipedia using the second method

Clustering is an unsupervised machine learning technique that groups together objects based on a similarity measure between them. This technique is appropriate for pairing documents in a comparable corpus as to their topic similarity. Classical document similarity measures rely on the supposition that the documents have common elements (words). But documents in different languages have actually very few common elements (numbers, formulae, punctuation marks, etc.) and in order to make documents in different languages similar, one approach is to replace the document terms with their equivalent translation pairs. In this approach, each document term is replaced with the translation equivalents pairs from a translation equivalents list. The document vectors for both source and target language documents are collections of translation equivalents pairs. There are several difficulties in this approach that have to be surpassed:

1. TRANSLATION EQUIVALENTS SELECTION. Not all the translation equivalents pairs have the same discriminative degree in differentiating between comparable documents.
2. CLUSTERING ALGORITHM MODIFICATIONS. The algorithm should consider pairing only different language documents.

TRANSLATION EQUIVALENTS TABLE. The accuracy of the comparable documents selection depends directly on the quality of the translation equivalents table. The translation equivalents table contains only content-word translations of lemmas with *N*-gram maximum lengths. Considering the fact that not all the translation equivalents have the same discriminative degree for selecting comparable documents, the translation equivalents table was filtered using a maximum translation equivalents entropy threshold (0.5 in our case). Using this filtering method, light verbs, nouns with many synonyms, and other spurious translation equivalents are removed.

DOCUMENT COLLECTION. The documents were tagged and lemmatized. Considering only the content words, for each n-gram from the document collection a set of translation equivalents were selected from the translation equivalents table. For example, the translation equivalents for "acetic acid" in both English and Romanian are: "acetic - acetic", "acetic acid - acid acetic", "acid - acid".

CLUSTERING FOR COMPARABLE DOCUMENTS IDENTIFICATION. This technique relies on the supposition that translation equivalents can be used as common elements that would make documents in different languages similar. We choose an agglomerative clustering algorithm. We tested several simple distance measures like Euclidean distance, squared Euclidean distance, Manhattan distance and percent disagreement. We found that percent disagreement differentiates better comparable and non-comparable documents. Considering the document vectors $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ of which elements are 1 or 0 depending on whether the corresponding vocabulary term belongs to the document or not, the percent disagreement is computed as:

$$d(x, y) = \frac{\sum_{i=1}^{n} x_i \neq y_i}{n}$$

The distance measure has the restriction that the compared documents have to be in different languages. This simple distance measure gave us a precision of 72% (with a maximum translation equivalents entropy threshold of 0.5 and a maximum of 3 translation equivalents per document term) on the collection of 128 English and Romanian Wikipedia documents described above.

Collecting Comparable Corpora from the Web

Data collection from the web is rarely a well defined job and more often than not corpus linguistics practitioners are designing their own scripts that provide an answer to the immediate need and as the problem is solved, the scripts are forgotten. Command line tools are usually applications designed to be used via text-only computer interface. We tried to give a more principled solution to reusing the small pieces of useful software and prolonging the life-time of such scripts. To this end, we developed an environment that incorporates three components: a Flow Graphical Editor which enables the user to easily

create and manage workflows, a Script Editor which assists the user in defining the processing units of the workflows and a Windows Service which takes as input the chained scripts generated by the first two components and executes the entire process at a given interval. As such, the environment is not a standalone crawler but a more general program which gives the means for high scalability and integration of modules written in different programming languages, interpreters or the use of the internal script developing system.

Out of the components described above, The Flow Graphical Editor component is the most important because it gives the advantage of graphically organizing the logic of the application around processing units and decision blocks. The user can alter the global application behavior by adding new blocks or modifying the way the output is being handled. One starts by creating the basic workflow. There are two types of active blocks: *decision blocks* and *processing units*. The Flow Graphical Editor allows for the integration of existing modules that produce console output, but the system can also enable the usage of other application types.

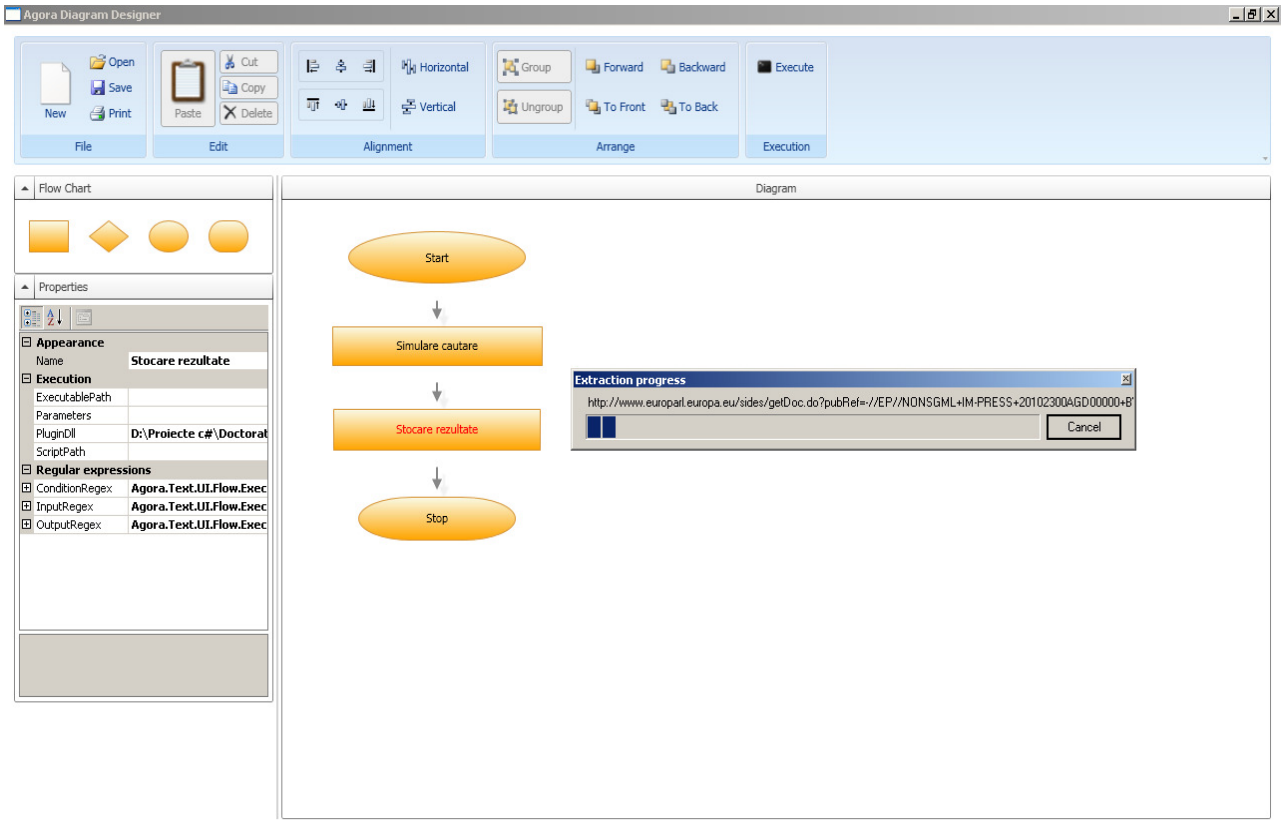

Figure 1: The Flow Graphical Editor and the execution of a diagram (a work flow)

As an example of orchestrated web crawling, we created a simple example in order to simulate the process and extract data from the European Parliament news archive. The European Parliament website provides a news section with an attached archive dating since 2004. The articles are translated 22 languages and are available for general use. The articles are classified in the sections and subsections and in order to retrieve specific articles classified accordingly, one has to perform a search using the European Parliament web interface and select its output which contains links to the desired articles.

The work flow (or diagram) for this example contains two processing units (see Figure 1): the first processing unit creates a list of articles by invoking the European Parliament web interface for searching articles in a given section and the second processing unit implements the actual data extraction which means downloading and storing the articles (provided by the first processing unit as links) on disk. At this point we can imagine a new processing unit which would feed from the output of the data extractor and would process the actual documents using the TTL web service (Tufiş et al., 2008). The design of the environment provides for this increased flexibility. If there is a need to crawl another website, one has to modify the script of the first processing unit that is responsible with collecting the links of the articles and the whole crawler is ready.

Conclusions

Mining the Web for MCC is an effective way of compensating the insufficient parallel corpora and there is a variety of different comparability levels that can be considered. Our aim is to collect MCC to enrich existing translation models. That is, we aim at extracting translation phrases (in addition to translation equivalents) from strongly MCC. To this end, we implemented several methods for strongly MCC acquisition that provided us with tens of millions of words worth of corpora. Also, we have developed a method for cross-lingually pairing documents enabling us to use the search engine gathering mechanism in order to collect strongly MCC.

Acknowledgements

The reported work has been carried within the ACCURAT project funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under the Grant Agreement n° 248347.

References

Moore, R. C. (2002). **Fast and Accurate Sentence Alignment of Bilingual Corpora**. In Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users, pages 135–144, London, UK, 2002. Springer-Verlag

Munteanu, D. Ş., and Marcu, D. (2006). **Extracting Parallel Sub-Sentential Fragments from Non-Parallel Corpora**. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 81–88, Sydney, July 2006. ©2006 Association for Computational Linguistics

Munteanu, D. Ş. (2006). **Exploiting Comparable Corpora**. PhD Thesis, University of Southern California, December 2006. ©2007 ProQuest Information and Learning Company

Skadiņa, I., Vasiļjevs, A., Skadiņš, R., Gaizauskas, R., Tufiş, D. Gornostay, T.: **Analysis and Evaluation of Comparable Corpora for Under Resourced Areas of Machine Translation**. In Proceedings of the 3rd Workshop on Building and Using Comparable Corpora, LREC 2010, Malta, pp. 6-14.

Tao, T., and Zhai, CX. (2005). **Mining Comparable Bilingual Text Corpora for Cross-Language Information Integration**. In Proceedings of the 2005 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005.

Tufiş, D., Ion, R., Ceauşu, Al., and Ştefănescu, D. (2008). **RACAI's Linguistic Web Services**. In Proceedings of the 6th Language Resources and Evaluation Conference - LREC 2008, Marrakech, Morocco, May 2008. ELRA - European Language Ressources Association. ISBN 2-9517408-4-0.

Yu, K., and Tsujii, J. (2009). **Extracting Bilingual Dictionary from Comparable Corpora with Dependency Heterogeneity**. In Proceedings of NAACL HLT 2009: Short Papers, pages 121–124, Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

Vu, T., Aw, A. T., and Zhang, M. (2009). **Feature-based Method for Document Alignment in Comparable News Corpora**. In Proceedings of the 12th Conference of the European Chapter of the ACL, pages 843–851, Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics