ACCURAT

Analysis and Evaluation of Comparable Corpora
for Under Resourced Areas of Machine Translation

www.accurat-project.eu

Project no. 248347

Deliverable D2.2
Report on multi-level alignment of comparable corpora

Version No. 1.0
31/08/2011

Document Information

Deliverable number:	D2.2
Deliverable title:	Report on multi-level alignment of comparable corpora
Due date of deliverable:	31/08/2011
Actual submission date of deliverable:	31/08/2011
Main Author(s):	Radu Ion, Xiaojun Zhang, Fangzhong Su, Monica Paramita, Dan Ştefănescu
Participants:	Radu Ion, Xiaojun Zhang, Fangzhong Su, Monica Paramita, Dan Ştefănescu
Internal reviewer:	Gregor Thurmair
Work package:	WP2
Work package title:	Multi-level alignment methods and information extraction from comparable corpora
Work package leader:	RACAI
Dissemination Level:	**PU**
Version:	V1.0
Keywords:	Document alignment, phrase alignment, sentence alignment, paragraph alignment, dictionary extraction, translation models

History of Versions

Version	**Date**	**Status**	**Name of the Author (Partner)**	**Contributions**	**Description/ Approval Level**
V0.1	25/07/2011	Draft	RACAI	RACAI	TOC
V0.2	25/07/2011	Draft	RACAI	RACAI	Added Introduction and Related Work @ Document Alignment
V0.3	26/07/2011	Draft	RACAI	RACAI	Added EMACC algorithm description and experiments
V0.4	27/07/2011	Draft	RACAI	RACAI	Added EMACC complete results in Annex 1 and 2
V0.5	10/08/2011	Draft	RACAI	RACAI	Formatted text and added PEXACC experiments
V0.6	11/08/2011	Draft	RACAI	USFD, CTS	Added

					contributions from partners
V0.7	22/08/2011	Draft	RACAI	RACAI	Added PEXACC description and addressed reviewer's comments
V0.8	26/08/2011	Draft	RACAI	RACAI	Minor corrections and references check.
V0.9	30/08/2011	Final	RACAI	DFKI	Added contribution from partners and modified common parts to include the new material.
V1.0	31/08/2011	Final	Tilde	Tilde	Submitted to EC

EXECUTIVE SUMMARY

This deliverable presents the methods of parallel data mining from comparable corpora developed at the time of writing within the ACCURAT project. It encompasses three algorithms of document alignment (EM based, SVM based and cosine similarity based) and two algorithms of parallel sentence/phrase extraction from paired comparable documents. The EM-based document aligner (called EMACC) and the parallel sentence/phrase extractor (called PEXACC) are thoroughly tested with complete evaluations presented in the annexes and are expected to be extensively used in the activity of gathering parallel data from comparable corpora.

Table of Contents

List of Tables

Abbreviations

Table 1 Abbreviations used throughout this document.

Abbreviation	Term/definition
EM	Expectation Maximization
ME	Maximum Entropy
SVM	Support Vector Machine
WP	Work Package
SMT	Statistical Machine Translation

Introduction

Statistical Machine Translation (SMT) is in a constant need of good quality training data both for translation models and for the language models. Regarding the latter, monolingual corpora are evidently easier to collect than parallel corpora and the truth of this statement is even more obvious when it comes to pairs of languages other than those both widely spoken and computationally well-treated around the world such as English, Spanish, French or German.

Comparable corpora came as a possible solution to the problem of scarcity of parallel corpora with the promise that it may serve as a seed for parallel data extraction. A general definition of comparability that we find operational is given by Munteanu and Marcu (2005). 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*.

Current practices of automatically collecting domain-dependent bilingual comparable corpora from the Web usually begin with collecting a list of t terms as seed data in both the source and the target languages. Each term (in each language) is then queried on the most popular search engine and the first N document hits are retained. The final corpus will contain $t \times N$ documents in each language and in subsequent usage the document boundaries are often disregarded.

At this point, it is important to stress the importance of the pairing of documents in a comparable corpus. Suppose that we want to word-align a bilingual comparable corpus consisting of M documents per language, each with k words, using the IBM-1 word alignment algorithm (Brown et al., 1993). This algorithm searches for each source word, the target words that have a maximum translation probability with the source word. Aligning all the words in our corpus with no regard to document boundaries, would yield a time complexity of k^2M^2 operations. The alternative would be in finding a 1:p (with p a small positive integer, usually 1, 2 or 3) document assignment (a set of aligned document pairs) that would enforce the "no search outside the document boundary" condition when doing word alignment with the advantage of reducing the time complexity to k^2Mp operations. When M is large, the reduction may actually be vital to getting a result in a reasonable amount of time. The downside of this simplification is the loss of information: two documents may not be correctly aligned thus depriving the word-alignment algorithm of the part of the search space that would have contained the right alignments.

Word alignment may form the basis of the phrase alignment procedure which, in turn, is the basis of any statistical translation model (Koehn et al. 2003). A comparable corpus differs essentially from a parallel corpus by the fact that textual units do not follow a translation order that otherwise greatly reduces the word alignment search space in a parallel corpus. This fact entails a lot of extra computation time in order to be able to detect correct alignments due to the extended search space. This being said, the algorithms that work on comparable corpora will have to be parallelized in order to cope with the enlarged search space.

In WP 2 we are committed to develop document and phrase alignment methods that leverage existing principles and are usable in the realm of comparable corpora. Through deliverable D2.6, "Toolkit for multi-level alignment and information extraction from comparable corpora" we are also committed to provide ready-to-use implementations of these algorithms that have been tested on all ACCURAT pairs of languages.

1. Document Alignment

1.1. Related Work

This section will report on existing document alignment algorithms and also, on other types of textual unit alignments such as sentences and/or paragraphs. We do not distinguish the algorithms for document alignment from those on sentence and/or paragraph alignment because, in principle, the same algorithms may be applied to any textual unit alignment (albeit with different accuracies).

Document alignment and other types of textual unit alignment have been attempted in various situations involving extracting parallel data from comparable corpora. The first case study is offered by Munteanu and Marcu (2002). They align sentences in an English-French comparable corpus of 1.3M of words per language by *comparing suffix trees of the sentences*. Each sentence from each part of the corpus is encoded as a suffix tree which is a tree that stores each possible suffix of a string from the last character to the full string. The algorithm for sentence alignment proceeds as follows:

a) generalized suffix trees are constructed for sentences in the source language and for those in the target language: one tree per language which is the concatenation of all suffix trees of all sentences in that language;
b) the source tree is checked against the target tree to determine branches that match. Since the vocabulary is not the same (the branches contain words from different languages), an initial bilingual lexicon is used to determine the match.

Using this method, Munteanu and Marcu are able to detect correct sentence alignments with a **precision of 95%** (out of 100 human-judged and randomly selected sentences from the generated output). The running time of their algorithm is approximately 100 hours for 50000 sentences in each of the languages.

Another popular method of aligning sentences in a comparable corpus is by *classifying pairs of sentences as parallel or not parallel*. Munteanu and Marcu (2005) use a Maximum Entropy classifier for the job trained with the following features: sentence lengths and their differences and ratios, percentage of the words in a source sentence that have translations in a target sentence (translations are taken from pre-existing translation lexicons), the top three largest fertilities, length of the longest sequence of words that have translations, etc. The training data consisted of a small parallel corpus of 5000 sentences per language. Since the number of negative instances (5000^2 – 5000) is much larger than the number of positive ones (5000), the negative training instances were selected randomly out of instances that passed a certain word overlap filter (see the paper for details). The classifier **precision is around 97%** with a **recall of 40%** at the Chinese-English task and **around 95%** with a **recall of 41%** for the Arabic-English task.

Chen (1993) *employs an EM algorithm that will find a sentence alignment* in the parallel corpus which maximizes the translation probability for each sentence bead in the alignment. The translation probability to be maximized by the EM procedure considering each possible alignment A is given by:

$$P(\mathcal{E}, \mathcal{F}, \mathcal{A}) = p(L) \prod_{k=1}^{L} P\left(\left[E_p^k; F_p^k\right]\right)$$

The following notations were used: $\mathcal{E}$ is the English corpus (a sequence of English sentences), $\mathcal{F}$ is the French corpus, $[E_p^k; F_p^k]$ is a sentence bead (a pairing of *m* sentences in English with *n* sentences in French), $\mathcal{A} = ([E_p^1; F_p^1], \dots, [E_p^L; F_p^L])$ is the sentence alignment (a sequence of sentence beads) and $p(L)$ is the probability that an alignment contains *L* beads. The EM algorithm developed by Chen is similar in principle with the one we're about to describe but there are several key differences that will be pointed out. Its **accuracy is around 96%** and was computed indirectly by checking disagreement with the Brown sentence aligner (Brown et al., 1991) on randomly selected 500 disagreement cases.

Another notable method of sentence alignment from "very-non-parallel corpora" is described in the work of Fung and Cheung (2004). Their contribution to the problem of textual unit alignment resides in *devising a bootstrapping mechanism* in which, after an initial document pairing and consequent sentence alignment using a simple lexical overlapping similarity measure, the IBM-4 model (Brown et al., 1993) is employed to enrich the bilingual dictionary that is used by the similarity measure. The process is repeated until the set of identified aligned sentences does not grow anymore. **The precision of this method on English-Chinese sentence alignment is 65.7%** (out of the top 2500 identified pairs).

Tao and Zhai (2002) use vectors of *relative frequency* in documents to compute similarities between words (in the two languages) in terms of a *Pearson's correlation coefficient* variant:

$$r(x,y) = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2\right)\left(\sum_{i=1}^{n} y_i^2 - \frac{1}{n}(\sum_{i=1}^{n} y_i)^2\right)}}$$

where x_i and y_i are the relative (normalized) frequencies of the words *x* and *y* in corresponding sets of documents and *n* is the total number of such sets.

Using this measure they construct a similarity function between documents as:

$$s(d_1, d_2) = \sum_{x \in d_1, y \in d_2} r(x,y) p(x|d_1) p(y|d_2)$$

where *x* and *y* are words, and $p(x|d)$ is the probability of occurrence of the word *x* in document *d*. The total number of word pairs is reduced by using an entropy threshold. This expression is improved by consecutively adding product factors or replacing some under the sum. The best expression that came out of this process is:

$$s(d_1, d_2) = \sum_{x \in d_1, y \in d_2} IDF(x) IDF(y) r(x,y) BM25(x, d_1) BM25(y, d_2)$$

where $IDF(x)$ is the inverse document frequency of the word *x*, and BM25 is the famous Okapi-BM25 term frequency normalization:

$$BM25(x,d) = \frac{k_1 \#(x,d)}{\#(x,d) + k_1\left(1 - b + b\frac{|d|}{AvgDocLen}\right)}$$

where $\#(x,d)$ is the frequency of the word *x* in document *d*, $|d|$ is the length of *d* in words, *AvgDocLen* is the average document length and k_1 and *b* are constants (usually set at 1.2 and 0.75 respectively). Using the formulas above, Tao and Zhai reported **86% alignment precision** among the top 100 document pairs returned by their system.

Vu et al. (2009) improve the previous method by adding new features that help the process of linking the source and target documents more effectively. First, they use a Date-Window filter in order to lower the search space, on the assumption that documents referring to the same subject must be written around the same date. They report that using a 1-day window

size, 81.6% for English-Chinese and 80.3% for English-Malay of the golden document alignments are covered. The increase of the window size to 5 raises the figures to 96.6% and 95.6%, respectively. Second, in order to further reduce the search space, they use the so-called *Title-n-Content* filter, which is intended to exploit knowledge of the documents' contents. This filter functions on the basis of a score that favors alignment candidates where at least one of the title-words in the source document has its translation found in the content of the target document:

$$TNC(d_1, d_2) = \sum_{x_i \in T_1} TR(x_i, c_2) + \sum_{x_j \in T_2} TR(x_j, c_1)$$

where $TR(x,c)$ is 1 if the translation of the word x is in c, and 0 otherwise. Here, c_i are the contents of documents d_i , while T_i the set of title words of the two documents. The authors report that using this filter they considerable reduced the alignment candidates: 47.9% for English-Chinese and 26.3% for English-Malay.

Vu et al. also improve Tao and Zhai's best formula by adding a linguistic feature that involves the comparison of the translation of words within a particular term in one language, and the presence of these translations in the corresponding target language term. In the view of the authors, the concept of *term* refers to multi-word expressions that act like single lexical units. Furthermore, they replace Pearson's correlation coefficient with *Discrete Fourier Transform* to calculate the similarity score of two frequency distributions and employ a feature named the *Linguistic Independent Unit*. This is well used in literature and refers to the information that is spelled alike in different languages, like numbers, dates, currency symbols, etc. The reported results are better than those obtained by Tao and Zhai (2002) with **4.1% for English-Chinese** and **8% for English-Malay** and better than those obtain by Munteanu (2006) with **23.2% and 15.3%**.

Munteanu and Marcu (2002) and Munteanu (2006) use the Lemur information retrieval toolkit (Ogilvie and Callan, 2001) to identify the most probable document pairs having similar content. Accordingly, each document in the source language is translated word-for-word and turned into a query, which is run against the collection of target language documents. The authors keep the top K (20) results as the most probable pairings for the query document. This approach is designed to ensure a high recall rather than a high precision of the alignment, because the main objective is not the document alignment itself, but the extraction of corresponding word and phrase translation equivalents.

Last but not least, Montalvo et al. (2006) devised a method for *multilingual document clustering based on the identification of cognate named entities*. The documents were extracted from a comparable corpus of English-Spanish news. The underlying principle of this method is that, usually, the target documents are translated (via MT systems or bilingual dictionary) in part or wholly, in the source language and then, classical, vector-based clustering techniques are applied to determine the clusters of similar documents. Montalvo et al. used named entities and their cognates for this job and obtained an **accuracy of 90.97%** which measured the percent of correct pairs found in any formed cluster.

1.2. EMACC: An Expectation-Maximization Algorithm for Textual Unit Alignment

We propose a specific instantiation of the well-known EM algorithm for aligning different types of textual units: documents, paragraphs, and sentences which we will name EMACC (an acronym for "Expectation Maximization Alignment for Comparable Corpora"). We draw our inspiration from the famous IBM models (specifically from the IBM-1 model) for word alignment (Brown et al., 1993) where the translation probability (eq. (5)) is modelled through

an EM algorithm where the hidden variable **a** models the assignment (1:1 word alignments) from the French sequence of words (‘ indexes) to the English one.

By analogy, we imagined that between two sets of documents (from now on, we will refer to documents as our textual units but what we present here is equally applicable – but with different performance penalties – to paragraphs and/or sentences) – let’s call them $\mathbf{E}$ and $\mathbf{F}$, there is an assignment (a sequence of 1:1 document correspondences[1]), the distribution of which can be modelled by a hidden variable z taking values in the set {true, false}. This assignment will be largely determined by the existence of word translations between a pair of documents, translations that can differentiate between one another in their ability to indicate a correct document alignment versus an incorrect one. In other words, we hypothesize that there are certain pairs of translation equivalents that are better indicators of a correct document correspondence than other translation equivalents pairs.

1.2.1. The Algorithm

We take the general formulation and derivation of the EM optimization problem from (Borman, 2009). The general goal is to optimize $P(X|\Theta)$, that is to find the parameter(s) Θ for which $P(X|\Theta)$ is maximum. In a sequence of derivations that we are not going to repeat here, the EM is given by:

$$\Theta_{n+1} = \underset{\Theta}{\operatorname{argmax}} \sum_{z} P(z|X,\Theta_n) \ln P(X,z|\Theta) \quad (1)$$

where $\sum_z P(z|X,\Theta_n) = 1$. At step *n*+1, we try to obtain a new set of parameters Θ_{n+1} that is going to maximize (the maximization step) the sum over z (the expectation step) that in its turn depends on the best set of parameters Θ_n obtained at step *n*. Thus, in principle, the algorithm should iterate over a set of parameters, compute the expectation expression for each of these parameters and choose the parameters for which the expression has the largest value. But as we will see, in practice, the set of all possible parameters has a dimension that is exponential in terms of the number of parameters. This renders the problem intractable and one should back off to heuristic searches in order to find a near-optimal solution.

Having the equation of the EM algorithm, the next task is to tailor it to the problem at hand: document alignment. But before doing so, let’s introduce a few notations that we will operate with:

- $\mathbf{E}$ is the set of source documents, $|\mathbf{E}|$ is the cardinal of this set;
- $\mathbf{F}$ is the set of target documents with $|\mathbf{F}|$ its cardinal;
- d_{ij} is a pair of documents, $d_i \in \mathbf{E}$ and $d_j \in \mathbf{F}$;
- $w_{ij}$is a pair of translation equivalents $\langle w_i, w_j \rangle$ such that w_i is a lexical item that belongs to d_i and w_j is a lexical item that belongs to d_j;
- $\mathbf{T}$ is the set of all existing translation equivalents pairs $\langle w_{ij}, p \rangle$. p is the translation probability score (as the one given for instance by GIZA++ (Gao and Vogel, 2008)). We assume that GIZA++ translation lexicons already exist for the pair of languages of interest.

In order to tie equation 1 to our problem, we define its variables as follows:

- Θ is the sequence of 1:1 document alignments of the form$D_{i_1 j_1}, D_{i_2 j_2}, \ldots, D_{ij} \in \{d_{ij} | d_i \in \mathbf{E}, d_j \in \mathbf{F}\}$. We call Θ *an assignment* which is basically a sequence of 1:1

[1] Or “alignments” or “pairs”. These terms will be used with the same meaning throughout the presentation.

document alignments. If there are $|\mathbf{E}|$ 1:1 document alignments in Θ and if $|\mathbf{E}| \leq |\mathbf{F}|$, then the set of all possible assignments has the cardinal equal to $|\mathbf{E}|!\binom{|\mathbf{F}|}{|\mathbf{E}|}$ where $n!$ is the factorial function of the integer n and $\binom{n}{k}$ is the binomial coefficient. It is clear now that with this kind of dimension of the set of all possible assignments, we cannot simply iterate over it in order to choose the assignment that maximizes the expectation;

- $z \in \{\text{true,false}\}$ is the hidden variable that signals if a pair of documents d_{ij} represents a correct alignment (true) or not (false);
- X is the sequence of translation equivalents pairs W_{ij} from **T** in the order they appear in each document pair from Θ.

Having defined the variables in equation 1 this way, our job is then to maximize $P(X|\Theta)$ meaning that we want to maximize the translation equivalents probability over a given assignment. In doing so, through the use of the hidden variable *z*, we are also able to find the 1:1 document alignments that attest for this maximization.

We proceed by reducing equation 1 to a form that is readily amenable to software coding. That is, we aim at obtaining some distinct probability tables that are going to be (re-) estimated by the EM procedure. Throughout the presentation, we will make some (overt and emphasized) independence assumptions that may or may not be correct but that are necessary in order to obtain the desired simplification. We acknowledge the fact that other derivations based on different assumptions are also possible.

We begin by expanding the expectation expression from equation 1

$$\sum_z P(z|X,\Theta_n)\ln P(X,z|\Theta) = \sum_z \frac{P(X,\Theta_n|z)P(z)}{P(X,\Theta_n)}\ln P(X,z|\Theta)$$

and making our first assumptions:

(**A1**) $P(X,\Theta_n) = P(X)P(\Theta_n)$

(**A2**) $P(X,\Theta_n|z) = P(X|z)P(\Theta_n|z)$

(**A3**) $P(X|z) = P(X)$

The third assumption (**A3**) states that X does not depend on z which only makes sense in the context of a document pair. The first assumption (**A1**) mandates that X and Θ_n are independent, which is justifiable if we think that X only depends on current Θ and not on the previously estimated one. The second assumption (**A2**) extends the first one by also imposing the same independence condition but conditioned on z. With these expressions at hand we proceed with the simplifications:

$$\sum_z \frac{P(X,\Theta_n|z)P(z)}{P(X,\Theta_n)}\ln P(X,z|\Theta)$$

$$= \sum_z \frac{P(X)P(z|\Theta_n)P(\Theta_n)P(z)}{P(X)P(\Theta_n)P(z)}\ln P(X,z|\Theta) = \sum_z P(z|\Theta_n)\ln P(X,z|\Theta)$$

A final assumption (**A4**) that we make is that $P(X,z|\Theta) = P(X|\Theta)P(z|\Theta)$or otherwise said, X and z are conditionally independent given Θ because z only makes sense in the presence of a document pair:

$$\sum_z P(z|\Theta_n)\ln P(X,z|\Theta) = \sum_z P(z|\Theta_n)\ln\big(P(X|\Theta)P(z|\Theta)\big)$$
$$= \ln P(X|\Theta)\sum_z P(z|\Theta_n) + \sum_z P(z|\Theta_n)\ln P(z|\Theta)$$

We thus end up with two probability tables: $P(X|\Theta)$which we call the lexical (document) alignment probability and $P(z|\Theta)$ which is the estimated assignment probability. But because of the fact that $\sum_z P(z|\Theta_n) = 1$ and being only interested in the $z = \text{true}$ value, we end up with the following simplified EM equation:

$$\Theta_{n+1} = \underset{\Theta}{\text{argmax}}[\ln P(X|\Theta) + P(\text{true}|\Theta_n)\ln P(\text{true}|\Theta)]$$

The probability $P(true|\Theta_n)$ is going to be a constant because it is computed based on the fixed assignment that was found in the previous step and thus, the previous equation is equivalent with maximizing the new EM

$$\Theta_{n+1} = \underset{\Theta}{\text{argmax}}[\ln P(X|\Theta) + \ln P(\text{true}|\Theta)] \quad (2)$$

Equation 2 suggests a method of updating the assignment probability $P(\text{true}|\Theta)$ with the lexical alignment probability $P(X|\Theta)$ in an effort to provide the alignment clues that will "guide" the assignment probability towards the correct assignment. All that remains to do now is to define the two probabilities according to our setup: document pairs and translation equivalents pairs.

The **lexical document alignment probability** $P(X|\Theta)$ is defined as follows:

$$P(X|\Theta) = \prod_{d_{ab}\in\Theta} \frac{\sum_{w_{ij}\in X} P(d_{ab}|w_{ij})}{|\mathbf{E}||\mathbf{F}|} \quad (3)$$

where $P(d_{ab}|w_{ij})$ is the simplified lexical document alignment probability which is initially equal to $P(w_{ij})$ from the set $\mathbf{T}$. This probability is to be read as "the contribution w_{ij} makes to the correctness of the d_{ab} alignment". We want that the alignment contribution of one translation equivalents pair w_{ij} to distribute over the set of all possible document pairs thus enforcing

$$\sum_{d_{ab}\in\{d_{xy}|d_x\in\mathbf{E},\, d_y\in\mathbf{F}\}} P\big(d_{ab}\big|w_{ij}\big) = 1 \quad (4)$$

The summation over X in equation 3 is actually over all translation equivalents pairs that are to be found only in the current d_{ab} document pair and the presence of the product $|\mathbf{E}||\mathbf{F}|$ ensures that we still have a probability value.

The **assignment probability** $P(\text{true}|\Theta)$ is also defined in the following way:

$$P(\text{true}|\Theta) = \prod_{d_{ab}\in\Theta} P(d_{ab}|\text{true}) \quad (5)$$

for which we enforce the condition:

$$\sum_{d_{ab}\in\{d_{xy}|d_x\in\mathbf{E},\, d_y\in\mathbf{F}\}} P(d_{ab}|\text{true}) = 1 \quad (6)$$

Using equations 2, 3 and 5 we deduce the final EM equation:

$$\Theta_{n+1} = \underset{\Theta}{\operatorname{argmax}} \left[\ln \prod_{d_{ab} \in \Theta} \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|\mathbf{E}||\mathbf{F}|} + \ln \prod_{d_{ab} \in \Theta} P(d_{ab}|\text{true}) \right]$$
$$= \underset{\Theta}{\operatorname{argmax}} \sum_{d_{ab} \in \Theta} \left[\ln \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|\mathbf{E}||\mathbf{F}|} + \ln P(d_{ab}|\text{true}) \right] \quad (7)$$

As it is, equation 7 suggests an exhaustive search *in the set of all possible Θ parameters*, in order to find the parameter(s) for which the expression that is the argument of "argmax" is maximum. But, as we already know, the size of this this set is prohibitive to the attempt of enumerating each Θ assignment and computing the expectation expression. Our quick solution to this problem was to directly construct the "best" Θ assignment[2] using a *greedy algorithm*: simply iterate over all possible 1:1 document pairs and for each document pair $d_{ab} \in \{d_{xy} | d_x \in \mathbf{E},\ d_y \in \mathbf{F}\}$, compute the alignment count (it's not a probability so we call it a "count" following IBM-1 model's terminology)

$$\ln \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|\mathbf{E}||\mathbf{F}|} + \ln P(d_{ab}|\text{true})$$

Then, construct the best 1:1 assignment Θ_{n+1} by choosing those pairs d_{ab} for which we have counts with the maximum values. Before this cycle (which is the basic EM cycle) is resumed, we perform the following updates:

$$P(d_{ab}|\text{true}) \leftarrow P(d_{ab}|\text{true}) + \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|\mathbf{E}||\mathbf{F}|} \quad (7a)$$

$$P(d_{ab}|w_{ij}) \leftarrow \sum_{d_{xy} \in \Theta_{n+1}} P(d_{xy}|w_{ij}) \quad (7b)$$

and normalize the two probability tables with equations 6 and 4. The first update is to be interpreted as the contribution the lexical document alignment probability makes to the alignment probability. The second update equation aims at boosting the probability of a translation equivalent if and only if it is found in a pair of documents belonging to the best assignment so far. In this way, we hope that the updated translation equivalent will make a better contribution to the discovery of a correct document alignment that has not yet been discovered at step *n* + 1.

Before we start the EM iterations, we need to initialize the probability tables $P(d_{ab}|\text{true})$ and $P(d_{ab}|w_{ij})$. For the second table we used the GIZA++ scores that we have for the w_{ij} translation equivalents pairs and normalized the table with equation 4. For the first probability table we have (and tried) two choices:

- (**D1**) a uniform distribution: $\frac{1}{|\mathbf{E}||\mathbf{F}|}$;
- (**D2**) a lexical document alignment measure $L(d_{ab})$ (values between 0 and 1) that is computed directly from a pair of documents d_{ab} using the w_{ij} translation equivalents pairs from the dictionary $\mathbf{T}$:

[2] We did not attempt to find the mathematical maximum of the expression from equation 7 and we realize that the consequence of this choice and of the greedy search procedure is not finding the true optimum.

$$L(d_{ab}) = \\ = \frac{\sum_{w_i \text{ in } d_a} f_{d_a}(w_i) \sum_{w_j \text{ in } d_b} f_{d_b}(w_j)}{|d_a||d_b|} \qquad (8)$$

where $|d_a|$ is the number of words in document d_a and $f_{d_a}(w_i)$ is the frequency of word w_i in document d_a (please note that, according to section 3, w_{ij} *is not* a random pair of words, but a pair of *translation equivalents*). If every word in the source document has at least one translation (of a given threshold probability score) in the target document, then this measure is 1. We normalize the table initialized using this measure with equation 6.

The algorithm proposed by Chen (1993) is the most similar in terms of the principles behind the method and thus, we need to pinpoint the differences .The main difference between the algorithm described by Chen and ours is that the search procedure reported there is invalid for comparable corpora in which no pruning is available due to the nature of the corpus. A second very important difference is that Chen only relies on lexical alignment information, on the parallel nature of the corpus and on sentence lengths correlations while we add the probability of the whole assignment which, when initially set to the D2 distribution, produces a significant boost of the precision of the alignment.

For the sake of the presentation we have preserved the 1:1 alignment restriction but that restriction is not enforced anymore. In the current version of the algorithm, when we construct the greedy assignment we allow at most p target alignments (we usually set p between 3 and 5) to one source document.

1.2.2. Experiments and Results

The test data for document alignment was compiled from the corpora that was previously collected in the project and that is known to the project members as the ”Initial Comparable Corpora” or ICC for short (further information is given in the ACCURAT deliverable D3.1 “Initial Comparable Corpora”). It is important to know the fact that ICC contains all types of comparable corpora from parallel to weakly comparable documents but we classified document pairs in three classes: parallel (class name: **p**), strongly comparable (**cs**) and weakly comparable (**cw**). We have considered the following pairs of languages: English-Romanian (en-ro), English-Latvian (en-lv), English-Lithuanian (en-lt), English-Estonian (en-et), English-Slovene (en-sl) and English-Greek (en-el). For each pair of languages, ICC also contains a Gold Standard list of document alignments that were compiled by hand for testing purposes.

We trained GIZA++ translation lexicons for every language pair using the DGT-TM[3] corpus. The input texts were converted from their Unicode encoding to UTF-8 and were tokenized using a tokenizer web service described by Ceauşu (2009). Then, we applied a parallel version of GIZA++ (Gao and Vogel, 2008) that gave us the translation dictionaries of content words only (nouns, verbs, adjective and adverbs) at wordform level. For Romanian, Lithuanian, Latvian, Greek and English, we had lists of inflectional suffixes which we used to stem entries in respective dictionaries and processed documents. Slovene remained the only language which involved wordform level processing.

The accuracy of EMACC is influenced by three parameters whose values have been experimentally set:

[3] http://langtech.jrc.it/DGT-TM.html

- the threshold over which we use translation equivalents from the dictionary **T** for textual unit alignment; values for this threshold (let's name it **ThrGiza**) are from the ordered set {0.001,0.4,0.8};
- the threshold over which we decide to update the probabilities of translation equivalents with equation 7b; values for this threshold (named **ThrUpdate**) are from the same ordered set {0.001,0.4,0.8};
- the top **ThrOut**% alignments from the best assignment found by EMACC. This parameter will introduce precision and recall with the "perfect" value for recall equal to **ThrOut**%. Values for this parameter are from the set {0.3,0.7,1}.

We ran EMACC (10 EM steps) on every possible combination of these parameters for the pairs of languages in question on both initial distributions D1 and D2. For comparison, we also performed a baseline document alignment using the greedy algorithm of EMACC with equation 8 supplying the document similarity measure. The following 6 tables report a synthesis of the results we have obtained (Annex 1 gives the full results). We omit the results of EMACC with D1 initial distribution because the accuracy figures (both precision and recall) are always lower (10-20%) than those of EMACC with D2.

Table 2 EMACC with D2 initial distribution on parallel corpora

p	**P**/R	Prms.	P/**R**	Prms.	#
en-ro	**1**/0.66666	* * <	1/**1**	0.4 0.001 1	21
en-sl	**0.98742**/0.29511	0.001 * 0.3	0.89097/0.89097	0.001 0.001 1	532
en-el	**1**/1	< * *	1/**1**	< * 1	87
en-lt	**1**/0.29971	0.4 0.001 0.3	0.93371/**0.93371**	0.001 0.8 1	347
en-lv	**1**/1	* * <	1/**1**	0.4 < 1	184
en-et	**1**/0.69780	* * 0.3	0.96153/0.96153	0.001 0.4 1	182

Table 3 D2 baseline algorithm on parallel corpora

p	**P**/R	Prms.	P/**R**	Prms.	#
en-ro	**1**/0.66666	* <	1/**1**	0.8 1	21
en-sl	0.98382/0.68738	0.001 0.7	0.93785/**0.93785**	0.001 1	532
en-el	**1**/0.69411	* <	1/**1**	0.001 1	87
en-lt	0.95192/0.28530	0.4 0.3	0.90778/0.90778	0.001 1	347
en-lv	**1**/0.29891	< 0.3	0.97826/0.97826	< 1	184
en-et	**1**/0.69780	< <	0.97802/**0.97802**	0.001 1	182

Table 4 EMACC with D2 initial distribution on strongly comparable corpora

cs	**P**/R	Prms.	P/**R**	Prms.	#
en-ro	**1**/0.69047	> * <	0.85714/**0.85714**	0.4 > 1	42
en-sl	0.96666/0.28807	0.4 0.4 0.3	0.83112/**0.83112**	0.4 0.4 1	302
en-el	**0.97540**/0.29238	0.001 0.8 0.3	0.80098/**0.80098**	0.001 0.4 1	407
en-lt	**0.97368**/0.29191	0.4 0.8 0.3	0.72978/**0.72978**	0.4 0.4 1	507
en-lv	**0.95757**/0.28675	0.4 > 0.3	0.79854/0.79854	0.001 0.8 1	560
en-et	**0.88135**/0.26442	0.4 0.8 0.3	0.55182/0.55182	0.4 0.4 1	987

Table 5 D2 baseline algorithm on strongly comparable corpora

cs	**P**/R	Prms.	P/**R**	Prms.	#
en-ro	**1**/0.69047	> <	0.85714/**0.85714**	0.4 1	42
en-sl	**0.97777**/0.29139	0.001 0.3	0.81456/0.81456	0.4 0.1	302
en-el	0.94124/0.28148	0.001 0.3	0.71851/0.71851	0.001 1	407
en-lt	0.95364/0.28514	0.001 0.3	0.72673/0.72673	0.001 1	507
en-lv	0.91463/0.27322	0.001 0.3	0.80692/**0.80692**	0.001 1	560
en-et	0.87030/0.26100	0.4 0.3	0.57727/**0.57727**	0.4 1	987

Table 6 EMACC with D2 initial distribution on weakly comparable corpora

cw	**P**/R	Prms.	P/**R**	Prms.	#
en-ro	**1**/0.29411	0.4 0.001 0.3	0.66176/**0.66176**	0.4 0.001 1	68
en-sl	**0.73958**/0.22164	0.4 0.4 0.3	0.42767/**0.42767**	0.4 0.4 1	961
en-el	**0.15238**/0.04545	0.001 0.8 0.3	0.07670/**0.07670**	0.001 0.8 1	352
en-lt	0.55670/0.16615	0.4 0.8 0.3	0.28307/**0.28307**	0.4 0.8 1	325
en-lv	**0.23529**/0.07045	0.4 > 0.3	0.10176/**0.10176**	0.4 0.4 1	511
en-et	**0.59027**/0.17634	0.4 0.8 0.3	0.27800/**0.27800**	0.4 0.8 1	483

Table 7 D2 baseline algorithm on weakly comparable corpora

cw	**P**/R	Prms.	P/**R**	Prms.	#
en-ro	0.85/0.25	0.4 0.3	0.61764/0.61764	0.4 1	68
en-sl	0.65505/0.19624	0.4 0.3	0.39874/0.39874	0.4 1	961
en-el	0.11428/0.03428	0.4 0.3	0.06285/0.06285	0.4 1	352
en-lt	**0.60416**/0.18012	0.4 0.3	0.24844/0.24844	0.4 1	325
en-lv	0.13071/0.03921	0.4 0.3	0.09803/0.09803	0.4 1	511
en-et	0.48611/0.14522	0.001 0.3	0.25678/0.25678	0.4 1	483

In every table above, the **P**/R column gives the maximum precision and the associated recall EMACC was able to obtain for the corresponding pair of languages using the parameters (**Prms.**) from the next column. The P/**R** column gives the maximum recall with the associated precision that we obtained for that pair of languages. The # column contains the size of the test set: the number of documents in each language that have to be paired. The search space is # * # and the gold standard contains # pairs of human aligned document pairs.

The **Prms.** columns contain parameter settings for EMACC (see Tables 2, 4 and 6) and for the D2 baseline algorithm (Tables 3 5 and 7): in Tables 2, 4 and 6 values for `ThrGiza`, `ThrUpdate` and `ThrOut` are given from the top (of the cell) to the bottom and in Tables 2, 4 and 6 values of `ThrGiza` and `ThrOut` are also given from top to bottom (the `ThrUpdate` parameter is missing because the D2 baseline algorithm does not do re-estimation). For the sake of compactness of representation we used some thresholds interval placeholders which are: "<" for the first two values of a threshold, ">" for the last two values and "*" for all values of a threshold. For instance, in Table 4, we have obtained a precision of 1 and a recall of 0.69047 aligning 42 en-ro documents with any of the values of {0.4,0.8} for `ThrGiza`, any of the values {0.001,0.4,0.8} for `ThrUpdate` and any of the values {0.3,0.7} for the `ThrOut` threshold.

To ease comparison between EMACC and the D2 baseline for each type of corpora (strongly and weakly comparable), we greyed maximal values between the two: either the precision in the **P**/R column or the recall in the P/**R** column.

In case of parallel corpora (Tables 1 and 2), we see that the initial distribution can already correctly align the parallel documents leaving little room for improvement for EMACC. In the case of strongly comparable corpora (Tables 4 and 5), we see that the benefits of re-estimating the probabilities of the translation equivalents (based on which we judge document alignments) begin to emerge with precisions for all pairs of languages (except en-sl) being better than those obtained with the D2 baseline. But the real benefit of re-estimating the probabilities of translation equivalents along the EM procedure is visible from the comparison between Tables 6 and 7. Thus, in the case of weakly comparable corpora, in

which EMACC with the D2 distribution is clearly better than the baseline (with the only exception of en-lt precision), due to the significant decrease in the lexical overlap, the EM procedure is able to produce important alignment clues in the form of re-estimated (bigger) probabilities of translation equivalents that, otherwise, would have been ignored.

It is important to mention the fact that the results we obtained varied a lot with values of the parameters `ThrGiza` and `ThrUpdate`. We observed, for the majority of studied language pairs, that lowering the value for `ThrGiza` and/or `ThrUpdate` (0.1, 0.01, 0.001…), would negatively impact the performance of EMACC due to the fact of *introducing noise* in the initial computation of the D2 distribution and also on *re-estimating (increasing) probabilities for irrelevant translation equivalents*. At the other end, increasing the threshold for these parameters (0.8, 0.85, 0.9…) would also result in performance decreasing due to the fact that *too few translation equivalents (be they all correct) are not enough to pinpoint correct document alignments* since there are great chances for them to actually appear in all document pairs.

So, we have experimentally found that there is a certain balance between *the degree of correctness of translation equivalents* and *their ability to pinpoint correct document alignments*. In other words, the paradox resides in the fact that if a certain pair of translation equivalents is not correct but the respective words appear only in documents which correctly align to one another, that pair is very important to the alignment process. Conversely, if a pair of translation equivalents has a very high probability score (thus being correct) but appears in almost every possible pair of documents, that pair is not informative to the alignment process and must be excluded. We see now that the EMACC aims at finding the set of translation equivalents that is maximally informative with respect to the set of document alignments.

We have introduced the `ThrOut` parameter in order to have better precision. This parameter actually instructs EMACC to output only the top (according to the alignment score probability $P(d_{ab}|\text{true})$) `ThrOut`% of the document alignments it has found. This means that, if all are correct, the maximum recall can only be `ThrOut`%. But another important function of `ThrOut` is to restrict the translation equivalents re-estimation (equation 7b) for only the top `ThrOut`% alignments. In other words, only the probabilities of translation equivalents that are to be found in top `ThrOut`% best alignments in the current EM step are re-estimated. We introduced this restriction in order to confine translation equivalents probability re-estimation to correct document alignments found so far.

We have also tested EMACC on the task of paragraph alignment in order to asses to what degree the performance of the algorithm decreases if it has to deal with textual units containing fewer words (and thus less lexical translation overlapping chances). We learned that the algorithms still performs with high accuracy when aligns paragraphs of at most 50 words. Annex 2 gives a detailed report on the results we obtained.

Regarding the running time of EMACC, we can report that on a cluster with a total of 32 CPU cores (4 nodes) with 6-8 GB of RAM per node, the total running time is between 12h and 48h per language pair (about 2000 documents per language) depending on the setting of the various parameters.

1.3. A SVM Document Pair Classifier with Feature-induced Levels of Parallelism

A document pair classifier is developed in this project and is mostly used to evaluate the retrieved comparable documents in WP3. However, this classifier can also be exploited in document alignments, in particular to choose and align comparable documents from a set of (unaligned) documents in the corpora. Given every possible pairing of documents in the

corpora, the classifier can predict the comparability level of each given pairs, and therefore enabling a subset of document pairs to be chosen as comparable documents.

This classifier is trained on the Initial Comparable Corpora (ICC, further information is given in the ACCURAT deliverable D3.1 "Initial Comparable Corpora") and is implemented using Thorsten Joachims' SVM^{light} method[4]. It makes use of several features from the documents, which can be divided into two types: language independent features and language dependent features. The list of all features used in the classifier is shown in the table below.

Table 8 List of all features

Language Independent Features	Language Dependent Features
Document Length (without translation)	Document Length (with translation)
Inter-links Overlap	Term Frequency Overlap
Out-links Overlap	Stemmed Term Frequency Overlap
Image Links Overlap	TF-IDF Overlap
Image Links Filename Overlap	Word Bi-gram Frequency Overlap
URL Level Overlap	Word Tri-gram Frequency Overlap
URL Character Overlap	

Given a pair of documents in the ACCES metadata format, all the above features are extracted from this pair and the classifier will classify this document pair to the pre-defined comparability classes: parallel, strongly comparable and weakly comparable or not comparable. The full description of this classifier is given in the ACCURAT deliverable D2.6, "Toolkit for multi-level alignment and information extraction from comparable corpora".

1.4. A Comparability Metric for Comparable Corpora

We have proposed and implemented two different comparability metrics (denoted by Metric 1 and Metric 2) for comparable corpora. In Metric 1, first bi-lingual dictionaries were automatically generated by using GIZA++ for word alignment on large-scale parallel text collections (e.g. Europarl and JRC-Acquis corpora). Then, we applied a statistical approach (log-likelihood co-occurrence statistics) for keyword extraction from the comparable documents. Using the generated dictionaries, the keyword vectors in source language were then translated into target language. Finally the cosine similarity measured was applied to measure the comparability of comparable document pairs. An initial investigation about the effectiveness of this metric has been carried out via experiments and the evaluation results have been given in the ACCURAT deliverable D1.2, "Report on metrics of comparability and parallelism".

The performance of Metric 1 highly relies on the quality of the automatically generated dictionary and the keyword vector translation is a key step in the metric design. However, the publically available parallel corpora are either too small or domain specific (for example, Europarl focuses on European parliament proceedings, and JRC-Acquis focuses on legal documents), making it hard to generate good dictionaries with broad word coverage across various different domains. Therefore, in Metric 2, instead of using GIZA++ based dictionaries for word translation, we applied the available translation APIs (e.g., Google

[4] http://svmlight.joachims.org/

translator and Microsoft Bing translator) for document translation. Translating documents from resource-poor languages into resource-rich languages also allows us to make better use of various existing language processing resources (such as word tokenizer, sentence splitter, word stemmer and lemmatizer, and POS tagger). Such language processing resources are not usually publically available or accurate enough for under-resourced languages.

In Metric 2, apart from the overlapped lexical information, we also incorporate some other information into the metric design. This includes: number of sentences in a document, number of content words (adjectives, adverbs, nouns, verbs, proper nouns) in a document, keywords extracted by a simple TF*IDF measure and the named entities extracted by the Stanford NER toolkit. The intuition justifying taking these features into account is that, if two documents are more comparable, they should have a similar number of sentences, a similar number of content words, and more overlapping of keywords and named entities between them.

We also performed an initial evaluation of Metric 2 by using Initial Comparable Corpora (ICC) as a gold standard. This evaluation was carried out by comparing the comparability scores obtained from the metric to the manually assigned comparability labels in ICC. Overall, the experimental results showed that the comparability scores (SC) produced by this metric can well indicate the comparability levels (parallel, strongly-comparable, and weakly-comparable) of tested document pairs in ICC, namely SC(parallel)>SC(strongly-comparable)>SC(weakly-comparable).

In order to better explore the usefulness of the designed comparability metrics, we will investigate the impact of comparability metric in other ACCURAT tasks, such as parallel phrases extraction from comparable documents and machine translation.

2. Phrase Mapping

Phrase-based statistical translation models are among the most successful translation models that currently exist (Callison-Burch et al., 2010). Usually, phrases are extracted from parallel corpora by means of symmetrical word alignment and/or by phrase generation (Koehn et al. 2003). In our case, we have to exploit comparable corpora to find parallel phrases and section 2.2 reports on an algorithm we have devised for this purpose and for the ACCURAT specific challenges.

2.1. Related Work

Available parallel corpora are not necessary ready to be used or even enough (when size is concerned) when Statistical Machine Translation (SMT) systems need it. This is especially true when it comes to languages not frequently used over Internet (which is now the main source of data) or even when a SMT system is due to be adapted to a new domain of application (e.g. medicine, renewable energy, automotive domain, etc.)

Parallel textual unit extraction (phrases, sentences, paragraphs) from different degrees of comparable corpora has been attempted in order to provide a solution to this MT data acquisition bottleneck problem. A comparable corpus is essentially different from a parallel one in that there is no guarantee that the available translations in the target part of the corpus:

- have a particular order in relation to the source segments of text they are supposed to translate; this particularity of comparable texts invalidate the hypothesis that holds on parallel texts: there is a given window (an ordered sequence of text segments) in the source part of the corpus such that any translation of a unit in this window is to be found only in the equivalent target window (Brown et al., 1991);
- are indeed intended translations of the same material or are accidental translations found there by mere chance; here we can take the example of Wikipedia articles in a language other than English which are, usually, created by reusing (translating) a (significant) part of the English version vs. some documents that are collected from the Web e.g. by imposing the same domain (sports, news, etc.) and the same date restrictions. It is clear that, in the case of Wikipedia articles, we can speak of genuine translations (that the editors of the foreign language articles generated in order to write their version of the subject) vs. accidental translations we may find using a subjective definition of "comparability". We thus argue that the difficulty of finding parallel material (or, equivalently, the accuracy of a specialized extraction algorithm) varies with the type of comparable corpora.
- have a certain coverage (high or medium) of the source material; we should not take as granted the fact that we e.g. are able to find parallel sentences in a comparable corpus and if we cannot, the corpus is not comparable or useless. Depending of the parallelism degree, we may be able to find only sub-sentential parallel fragments, e.g. parallel noun phrases, verb phrases, named entities, terminologies, etc. But this commitment involves a high computing time equivalent to that of a brute force search where one should score every source fragment with every target fragment.

Previous methods of finding parallel material in comparable corpora have taken one of the following main roads:

- classify pairs of textual units (paragraphs or sentences) as parallel or not (Munteanu and Marcu, 2006);

- using a proper SMT system to translate source textual units into the target language and match the translation with existing target fragments in the target language (Abdul-Rauf and Schwenk, 2009);
- using different (translation-wise) similarity measures to match source and target textual units (Fung and Cheung, 2004). Our approach (to be described in section 2.2) also uses this general approach;
- a generative story: the target textual units are generated by translation from selected source textual units and a probabilistic model describing both alignment and translation has been proposed (Quirk et al., 2007).

General descriptions of the algorithms and their evaluations from (Munteanu and Marcu, 2006) and (Fung and Cheung, 2004) have already been given in section 1.1.

2.2. PEXACC: A Phrase Mapping Algorithm for Comparable Corpora with Relevance Feedback

We have developed an algorithm for parallel data mining from comparable corpora, which we will call PEXACC – the short from Parallel phrase EXtrActor from Comparable Corpora, that is tightly adapted to the needs of the ACCURAT project. The type of comparable corpora we initially needed to deal with (that is, when PEXACC concept was thought of) was that of a weakly comparable corpus:

- source language and target language documents were independently collected from the Web;
- broad domain restrictions were enforced: newswires;
- time restrictions were also enforced: documents from the same period of time were collected where the period of time was limited to a couple of days;
- the property of "presenting the same story" was formalized as having a number of named entities in common.

The resulting corpus was a weakly comparable corpus in which we experimentally found that translations appeared in the overwhelmingly vast majority of cases only on the sub-sentential level (noun phrases, named entities and some terminology). That being the case, PEXACC was structured from the very beginning to:

- be able to split the sentences of a document in smaller parts for which it became possible to find translations given the nature of comparable corpora it had to deal with (but it also retained the possibility to find parallel equivalents for entire sentences);
- implement an exhaustive search of all possible pairs of source/target text parts because any kind of anchoring was deemed irrelevant since there were no real parallel pieces of text larger than a couple of words.

PEXACC is a parallel phrase extractor that belongs to the category of extractors that score pairs of phrases or sentences according to some kind of lexical overlap and structural matching measure. Actually, PEXACC linearly combines a set of feature functions (which output translation similarity scores between 0 and 1) to obtain the final "parallelism" score *P* of two phrases *e* (in the source language) and *f* (in the target language):

$$P(e,f)=\sum_i w_i f_i(e,f), \quad \sum_i w_i = 1, \quad 0 \le f_i(e,f) \le 1 \ \ \forall e,f,i \qquad (1)$$

The feature functions are designed to return a value close to 1 if the arguments are parallel phrases/sentences so that the *p* value is close to 1 for parallel *e* and *f*.

The general workflow of PEXACC is as follows (given a pair of source and target documents):

1. split the input source and target documents into sentences and then, if desired, into smaller parts (loosely called 'phrases' throughout this presentation) according to a list of language dependent markers (we call it 'the PEXACC fragmentation routine'). By a "marker" we understand a specific functional word that, usually, indicates the beginning of a syntactic constituent or a clause. For English these markers include: prepositions, particles and negations (the infinitive 'to', 'not'), auxiliary and modal verbs ('have', 'be', 'can', 'must'), interrogative and relative pronouns, determiners and adverbs ('which', 'what', 'who', 'that', 'how', 'when', 'where', etc.) and subordinating conjunctions ('that', 'as', 'after', 'although', 'because', 'before', etc.). A very important design decision here is choosing a set of markers such that, for the source and the target languages, the text parts we obtain by splitting are in a 1:1 correspondence as much as possible. Thus, for Romanian, the same types of markers can be considered and, in most of the cases, the text parts would align 1:1 if the splitting process is applied on a parallel pair of sentences. In the example pair of parallel sentences (the markers are underlined, square brackets indicate the parts):

   en: `[A simple example] [will demonstrate the splitting] [of this sentence] [into smaller parts].`

   ro: `[Un exemplu elementar] [va demonstra împărţirea acestei propoziţii] [în părţi mai mici].`

   we have the following correspondences: "`[A simple example]` ⇔ `[Un exemplu elementar]`" (1:1 correspondence), "`[will demonstrate the splitting] [of this sentence]` ⇔ `[va demonstra împărţirea acestei propoziţii]`" (2:1) and "`[into smaller parts]` ⇔ `[în părţi mai mici]`" (1:1).
2. score each possible pair of text parts (sentences of phrases) *e* and *f* as to their parallelism degree by using equation 1;
3. output all pairs of text parts for which equation 1 gives a score larger than a predefined threshold (set to 0.1 but the real parallelism threshold is dependent on the type of the corpus: parallel, strongly comparable and weakly comparable – see section 2.2.1).

Equation 1 makes use of several feature functions that are designed to indicate the parallelism of two phrases *e* and *f*. These functions are designed to return 1 when *e* and *f* are perfectly parallel (i.e. *f* has been obtained from *e* by translation if *e* and *f* were to be presented together as a pair to a human judge). The functions should return a value close to 0 when *e* and *f* are not related at all but this behaviour is critically influenced by the quality and the completeness of the dictionary that is used. Thus, *e* and *f* may still be parallel but if individual words in *e* do not have the relevant *f* translations in the dictionary and/or the translations probabilities are small, the resulting (low) score could be misleading. This is the main reason for which we have incorporated a "**relevance feedback loop**" (idea from (Fung and Cheung, 2004)). Thus, the 4th step of the algorithm is executed for a fixed number of steps and

4. takes the output of step 3 and trains a supplementary GIZA++ dictionary on all text parts pairs with a certain parallelism score (to minimize noise) and adds it to the main initial dictionary. The combination method between the main dictionary *D* and the learnt one *T* is as follows:
   - if the pair of the translation equivalents *t* is found in both dictionaries its new translation probability $p(t)$ becomes $p(t) = 0.7 \times p_D(t) + 0.3 \times p_T(t)$ where

$p_D(t)$ is the probability of t in the D dictionary and $p_T(t)$ is the probability of t in the T dictionary;

- if the pair of translation equivalents t is found in either D or T but not both, leave its probability unchanged.

Each feature function from equation 1 is weighted according to the importance we attribute to the corresponding feature. These weights have been experimentally set but an optimization procedure may be applied to optimally determine the value of these weights according to some training data. The feature functions that are used by equation 1 are as follows (each corresponding weight is also given):

- $f_1(e, f)$ or the "**lexical (translation) overlap**" feature function ($w_1 = 0.6$). This function measures how many (and how well) source words from e are translated in f. Formally, if:
  - e_i and f_j are words from e and f (located at positions i and j respectively) such that f_j translates e_i according to the main (D) dictionary in a competitive linking manner (Melamed, 2001),
  - $L(e)$ is the length of e in words,
  - $p(e_i, f_j)$ is the translation probability of the pair $\langle e_i, f_j \rangle$ from D and
  - $N(e)$ is the number of words in e that have been translated in f according to the main dictionary D, then

$$f_1(e, f) = \left(\frac{N(e)}{L(e)} \right)^{\frac{L(e)}{N(e)}} \frac{\sum_{i,j} p(e_i, f_j)}{L(e)}$$

- $f_2(e, f)$ or the "**alignment locality**" feature function ($w_2 = 0.15$). This function is able to cumulatively evaluate if the relative indexes of the source and target words that align in e and f are not very different. In other words it measures the degree in which the alignments of source words in e "land" at similar relative indices in f assuming that the word order of e and f is not very different (e.g. true of English and Romanian). Formally, if:
  - A is the set of all pairs of indices $\langle i, j \rangle$ such that e_i and f_j are words from e and f (located at positions i and j respectively) and f_j translates e_i according to the main (D) dictionary in a competitive linking manner,
  - $L(e)$ and $L(f)$ are the lengths of e and f in words,
  - $\|A\|$ is the number of elements from the A set,
  - $|x|$ is the absolute value of x, then

$$f_2(e, f) = 1 - \frac{\sum_{\langle i,j \rangle \in A} \left| \frac{i}{L(e)} - \frac{j}{L(f)} \right|}{\|A\|}$$

- $f_3(e, f)$ is the "**both ends of the source phrase have translations**" feature function ($w_3 = 0.15$). This function returns 1 if at least one of the first couple of words (configurable parameter, set experimentally to 3) from e is translated in f by a word in the first couple of words and at least one of the last couple of words from e is also

translated in *f* by a word in last couple of words. It returns 0 in the opposite case. The intuition behind this function is that if two phrases are parallel, then their ends must match if the word order is usually preserved.

- $f_4(e, f)$ is the "**have the same entities present**" feature function ($w_4 = 0.09$). This function returns 0 if *e* contains some numerical and/or named entities and *f* does not or the vice versa. It returns 1 in the opposite case. In other words, we want that numerical and/or named entities in *e* to be echoed in *f*.
- $f_5(e, f)$ is the "**have the same punctuation at end**" feature function ($w_5 = 0.01$). This function returns 1 if *e* and *f* end with the same punctuation and 0 in the opposite case.

The parallelism score of PEXACC from equation 1 can be easily extended by adding new feature functions and making sure that they implement the same functionality: return a value close to 1 for parallel arguments and a value close to 0 for unrelated arguments. For instance, following the example from (Abdul-Rauf and Schwenk, 2009), one can incorporate an existing SMT system that will output an *f'* when given *e*. Then, the job of the new feature function would be to measure the similarity between *f* and *f'* monolingually.

PEXACC can also work with sentences instead of phrases. If one knows that the comparable corpus contains an important parallel part, then the algorithm can be configured to search for parallel sentences instead of parallel phrases. When mapping phrases, one of the main deficiencies of PEXACC is that it does not retain the position of the phrase in the document (either source or target). This way, consecutive phrases that have been successfully mapped (both in the source and in the target document) cannot be combined so that the source-adjoined phrase is directly aligned (not mapped since we have positional information) to the target-adjoined phrase. This will be the next step in the development of PEXACC.

2.2.1. Experiments and Results

The assumption on which we based our entire evaluation process is that *if PEXACC has a specific (measurable) accuracy on a (random) pair of parallel documents, that accuracy should not significantly degrade if we introduce noise (in quantifiable ratios to the existing parallel data) in the source and target documents and randomly permute the sentences in each document*. To test that assumption, we needed to construct a Gold Standard (GS) of mapped phrases from a pair of (clean) parallel documents and we needed to have such GSs for all pairs of languages in test. This last requirement made us to turn to an automatic method of constructing a GS (since we are not able to read in all the languages in test):

- given a reference pair of parallel documents (tested with 100 sentences per document, randomly selected from different domains; document pairs are the MT Test Data parallel document pairs of the ACCURAT project which exist for all project languages),
- apply GIZA++ to obtain a word alignment from the source sentences to the target sentences;
- for each word-aligned source sentence and target sentence pair, break them using PEXACC fragmentation routine (see the previous subsection) and align the resulting text fragments based on word alignments such that links of words from a source fragment do not point outside the boundaries of a target fragment.

For instance, given the English sentence "In addition to schools and universities, the drive is on for libraries, museums and similar institutions ..." and the Romanian translation "În plus faţă de şcoli şi universităţi, se acţionează pentru ca bibliotecile, muzeele şi instituţii similare

…", Figure 1 displays the PEXACC fragmentation style using dotted lines. Along with GIZA++ generated word alignments (see the arrows from the English words to the Romanian equivalents) we are able to automatically generate GS phrase mappings "In addition" ⇔ "În plus faţă", "to schools" ⇔ "de şcoli", "and universities," ⇔ "şi universităţi,", etc. The quest is on then to apply PEXACC onto the same pair of parallel documents but with added noise (random sentences added in the same proportion to the source document and the target document) and see to what extent we can cover the GS and with what precision we can generate parallel mapped phrases.

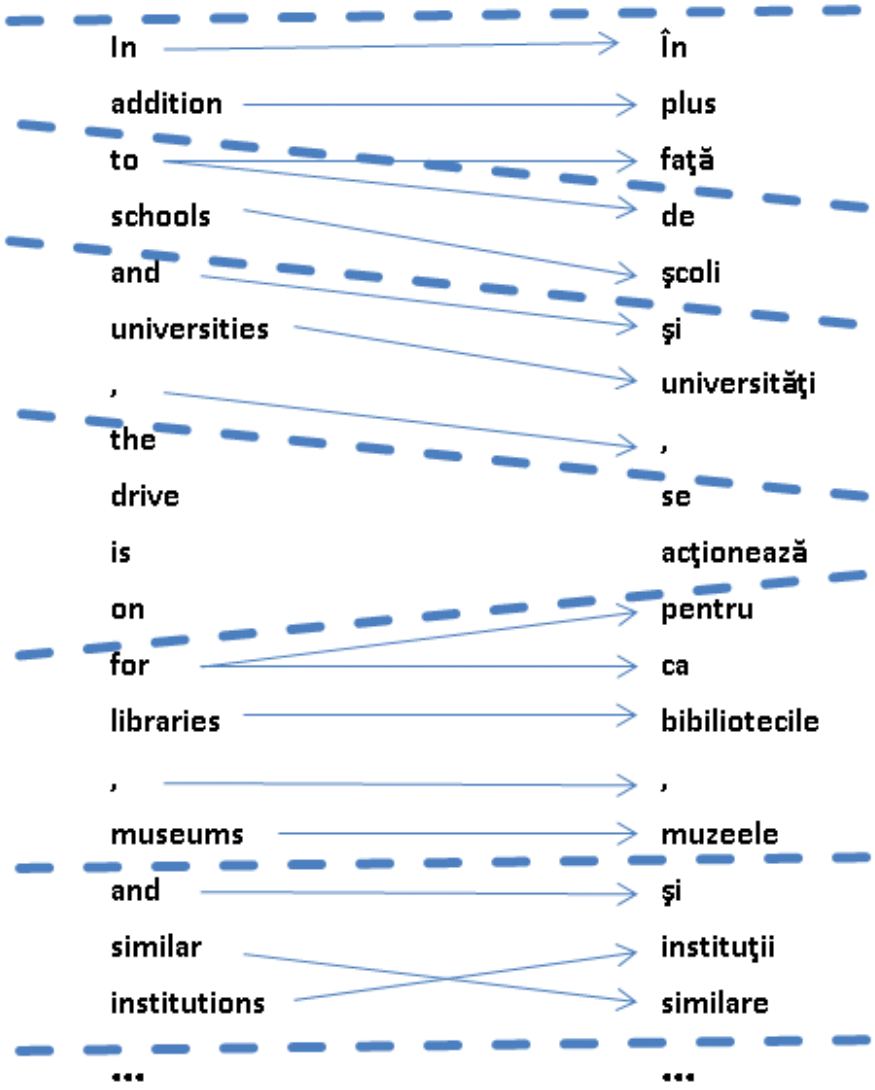

Figure 1: PEXACC fragmentation example in English and Romanian. GS will contain pairs of phrases delimited by the dotted lines and supported by GIZA++ generated word alignments (drawn as arrows from the English words to the Romanian equivalents).

Table 9 PEXACC performance on the parallel EN-RO document pair, using a JRC Acquis-based GIZA++ extracted dictionary; 3 relevance feedback loops (the maximal values for each category P, R and F are bolded).

	Iteration 1		Iteration 2		Iteration 3	
0.1	P: 0.53396 R: 0.75221 F: 0.62456	530 unique EN phrases	P: 0.54887 R: **0.76106** F: 0.63778	532 unique EN phrases	P: 0.55075 R: **0.76106** F: 0.63904	532 unique EN phrases
0.3	P: 0.63403 R: 0.70796 F: 0.66896	429 unique EN phrases	P: 0.64759 R: 0.72123 F: 0.68243	437 unique EN phrases	P: 0.64464 R: 0.72123 F: 0.68079	439 unique EN phrases
0.5	P:**1** R: 0.50884 F: 0.67448	228 unique EN phrases	P:**1** R: 0.51769 F: 0.68221	239 unique EN phrases	P:**1** R: 0.52654 F: **0.68985**	241 unique EN phrases

Table 10 PEXACC performance on the parallel EN-RO document pair, using a (very large) reference dictionary; 3 relevance feedback loops (the maximal values for each category P, R and F are bolded).

	Iteration 1		Iteration 2		Iteration 3	
0.1	P: 0.72280 R: **0.80973** F: 0.76380	570 unique EN phrases	P: 0.71278 R: **0.80973** F: 0.75817	571 unique EN phrases	P: 0.71278 R: **0.80973** F: 0.75817	571 unique EN phrases
0.3	P: 0.78160 R: 0.78318 F: 0.78239	522 unique EN phrases	P: 0.76615 R: 0.78761 F: 0.77673	526 unique EN phrases	P: 0.76136 R: 0.77876 F: 0.76996	528 unique EN phrases
0.5	P:**1** R: 0.69469 F: **0.81984**	396 unique EN phrases	P:**1** R: 0.67256 F: 0.80423	390 unique EN phrases	P:**1** R: 0.67699 F: 0.80738	391 unique EN phrases

We have to note the following deficiencies of this automatically generated GS:

- word-alignments generated by GIZA++ are not perfect and as such, there are correct phrase mappings that PEXACC finds but that are not present in the GS on the account that the supporting word-alignments were missing/wrong;
- the GS was generated from a pair of parallel documents that are word-aligned at sentence level. But PEXACC may also find correct phrase mappings with phrases belonging to sentences that not paired; these (correct) phrase mappings will obviously not be present in the GS. Thus, in order to compute a fair precision with respect to the

given GS, we are going to consider as the set of generated results, the set of all the source phrases[5] that PEXACC found. We give a precision point and a recall point to PEXACC if for a given source phrase, there is target phrase mapped to it such that the pair is found in GS. In addition to that, we also experimentally observed that, for English-Romanian, all phrase pairs with a parallelism probability of over 0.5 are in fact correct even if they are not found in the GS[6]. In this case, we will also give PEXACC a precision point (but not a recall point) if the detected phrase pair has at least 0.5 as its parallelism probability.

Tables 9 and 10 report on the base line performance of PEXACC: running on a pair of parallel clean documents that do not contain any added noise. Table 9 presents the run using an English-Romanian GIZA++ dictionary extracted from the JRC Acquis corpus[7] and Table 10 presents the same run but using a very large (over 9.5 million entries at wordform level) English to Romanian dictionary extracted from all our parallel corpora and enriched with a WordNet based dictionary derived from the conceptual alignments between the Princeton WordNet[8] and the Romanian WordNet (Tufiş et al., 2008). There are 3 parallelism thresholds for which we computed the precision (P), the recall (R) and the F-measure (F) of the algorithm: 0.1, 0.3 and 0.5. After each phrase extraction phase (called 'an iteration'), a GIZA++ dictionary is trained on the output of the algorithm (considering all pairs of phrases with a parallelism probability of at least 0.5) and the resulting dictionary is incorporated into the main dictionary. Before 'Iteration 1' we have only the main dictionary.

Studying the Tables 9 and 10 comparatively, we can observe the following facts:

- we can obviously improve the extraction accuracy by using a better (larger and more accurate) dictionary (see Table 10) but, in that case, training new dictionaries will not improve our subsequent extraction steps (in Table 10, the best result is obtained in the first iteration) due to the fact that the new translation equivalents pairs are very rare. This is the explanation of the fact that we cannot achieve 100% recall: no matter how large one dictionary is, it will always be incomplete with respect to new data. Figure 1 contains an example where the phrase "the drive is on" is the equivalent of the Romanian "se acţionează"; the translation pair "drive, acţionează" is a new translation pair missing from our huge dictionary;
- on the other hand, training intermediary GIZA++ dictionaries certainly helps to discover new translation pairs (see Table 9 where better results are obtained with each new iteration) when using a rather small (just over 200 thousand entries at wordform level) dictionary. Since we cannot rely on the existence of accurate and large dictionaries for every language pair, we need to adopt this "extract, learn and loop" strategy. This is the main reason for which all subsequent tests for all intended language pairs will use JRC-Acquis extracted dictionaries.

[5] And, if it is a set, no source phrase is repeated.

[6] The probability threshold over which all generated parallel pairs is correct is dependent on the type of document pairs. For the English-Romanian pair of parallel documents on which we tested, at least 0.5 is guaranteed to indicate perfect parallelism (we have determined that by manually inspecting the output).

[7] http://langtech.jrc.it/JRC-Acquis.html

[8] http://wordnet.princeton.edu/

Table 11 PEXACC performance on the strongly comparable EN-RO document pair (noise ratio 1:1), using a JRC Acquis-based GIZA++ extracted dictionary; 3 relevance feedback loops (the maximal values for each category P, R and F are bolded).

	Iteration 1		Iteration 2		Iteration 3	
0.1	P: 0.24594 R: 0.73008 F: 0.36794	1110 unique EN phrases	P: 0.27549 R: **0.73893** F: 0.40135	1118 unique EN phrases	P: 0.27638 R: **0.73893** F: 0.40229	1118 unique EN phrases
0.3	P: 0.29571 R: 0.68584 F: 0.41324	886 unique EN phrases	P: 0.32461 R: 0.69911 F: 0.44336	918 unique EN phrases	P: 0.32359 R: 0.69911 F: 0.44241	924 unique EN phrases
0.6	P: **1** R: 0.33185 F: 0.49833	185 unique EN phrases	P: **1** R: 0.39380 F: 0.56507	232 unique EN phrases	P:**1** R: 0.41150 F: **0.58307**	237 unique EN phrases

Table 11 contains the results of running PEXACC on our pair of parallel documents to which we have added (to each individual document in fact) noise in proportion of 1:1 meaning that for each existing sentence in the document, another random one was added (we have selected the random sentences from ICC). This noise addition modified the status of our document pair from 'parallel' to 'strongly comparable'. After the noise sentences were added, a random permutation of the sentences in each document was generated to ensure that the order in which the parallel sentences appear does not influence the outcome of PEXACC.

After running the phrase extractor tool on the modified documents we noticed that the parallelism probability above which all extracted pairs were correct (perfectly parallel) increased to 0.6. This happened due to the following facts:

- the extractor encountered pairs of phrases in which bad translation equivalents exist which, despite the fact that they do not have large translation probabilities, their number and disposition in each of the phrases in the pair fool the similarity measure;
- we did not differentiate between functional words and stop words when we considered pairs of translation equivalents that influence the similarity measure; thus many pairs in which only stop words are responsible for the large similarity measure exist. We will fix this behavior in future versions of PEXACC;

But although *all the extracted pairs over the 0.6 threshold* are in fact parallel, there are many pairs over 0.5 which are also perfect parallel pairs: "A new era" ⇔ "O nouă eră", score 0.58, "to the hospital." ⇔ "spre spital.", score 0.52, etc. But because these pairs do not exist in our initial GS, we have no means to count them as precision points. Finally, we have to stress the fact that many correct pairs over 0.6 still cannot be found in GS. With these considerations in mind one should judge the lower precision/recall of PEXACC on the noise-induced comparable pair of documents vs. the parallel pair of the same documents.

The important thing to notice about Table 11 is that the recall – when considering all the pairs over the lowest accepted parallelism probability of 0.1 – does not significantly decrease (a 2.2% decrease) when compared to the baseline in Table 9. This fact confirms that the only limitation of PEXACC in finding all relevant parallel pairs resides in the dictionary used and not in the order and/or amount of sentences in a document or the 'comparability' level of the

document pair. This finding is obvious if one thinks about how PEXACC actually works: *by trying all combinations of source and target phrases and score each combination individually*. It cannot skip a pair no matter how much noise one adds to each document in the pair. But it fails in other respect: the value of parallelism probability that indicates true parallelism does not stay the same when we go from parallel documents to comparable documents and to weakly comparable documents. This will be our main focus in future development of PEXACC.

Table 12 confirms the fact that the recall is not significantly affected with the addition of noise, once more. Here we ran PEXACC on a noise-altered version of our parallel document pair containing noise in a proportion of 2:1.

Table 12 PEXACC performance on the strongly comparable EN-RO document pair (noise ratio 2:1), using a JRC Acquis-based GIZA++ extracted dictionary; 3 relevance feedback loops (the maximal values for each category P, R and F are bolded).

	Iteration 1		Iteration 2		Iteration 3	
0.1	P: 0.14645 R: 0.73008 F: 0.24396	1509 unique EN phrases	P: 0.17049 R: **0.73893** F: 0.27705	1525 unique EN phrases	P: 0.16513 R: **0.73893** F: 0.26994	1526 unique EN phrases
0.3	P: 0.18072 R: 0.68584 F: 0.28606	1162 unique EN phrases	P: 0.20712 R: 0.69911 F: 0.31957	1207 unique EN phrases	P: 0.19835 R: 0.69469 F: 0.30859	1215 unique EN phrases
0.7	P:**1** R: 0.23451 F: 0.37992	110 unique EN phrases	P: **1** R: 0.26548 F: **0.41958**	154 unique EN phrases	P:**1** R: 0.26548 F: **0.41958**	154 unique EN phrases

2.3. Parallel Sentence Extraction Using Maximum Entropy Modeling

The main goal of our work is to extraction parallel sentence pairs from a comparable corpus. The prototype in (Munteanu and Marcu, 2005) is implemented: Firstly we list all sentence combinations from document-aligned Wikipedia corpus; then we use a GIZA lexicon to filter the candidate sentence that are possible to be parallel; finally a maximum entropy (ME) classifier is applied that classifies each candidate pair as 'parallel' or 'non-parallel'.

As Figure 2 shows, we divide the workflow into two parts: training (above the dash line) and extraction process (under the dash line).

The use of the training corpus

We use an initial corpus for providing lexicon translation probabilities and employ GIZA++ which runs with the standard configuration: 5 iterations of IBM-1 model, 3 iterations in both IBM-3 and IBM-4 models, 5 iterations in HMM model. The alignment process runs in both directions and then, we symmetrize the alignments using the refined heuristic. After that, a GIZA lexicon is trained and could be used as a resource for both sentence filtering and classification. In this lexicon table, one word t_i in source language may be aligned with

multiple words e_j in the target language; each pair is given a positive real value which indicates the conditional translation probability $p(e_j|t_i)$. The entries with high translation probabilities are correct while those with lower probabilities may still be correct. This translation lexicon is used to select candidate sentences.

Another utility of the initial corpus is to provide training data for ME classification. The parallel sentence pairs could be viewed as positive samples; negative samples are generated by scrambling the order of the sentences in the corpus.

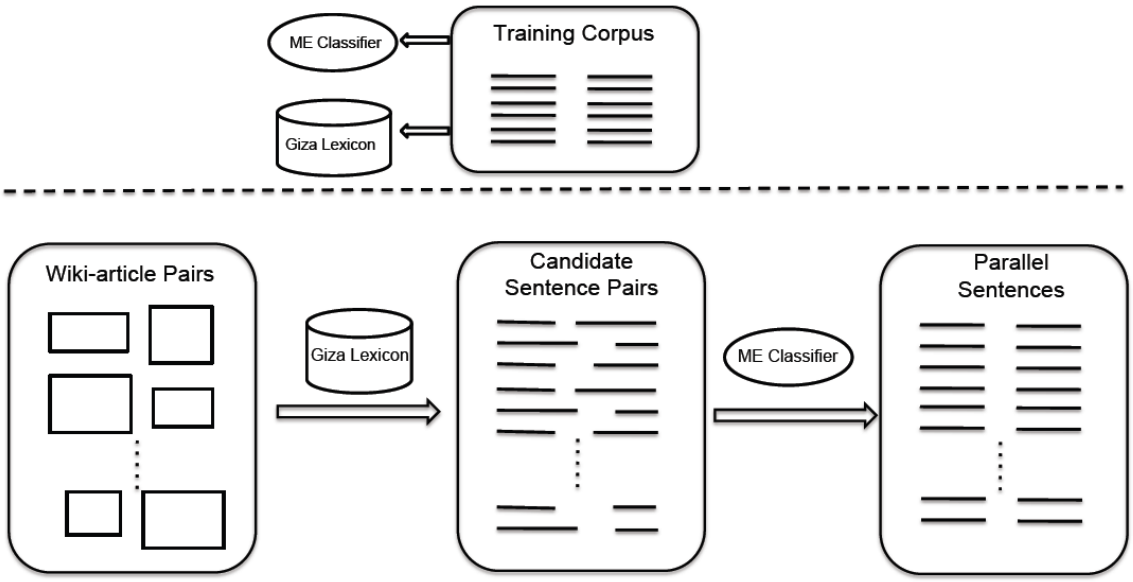

Figure 2: An overview of the parallel sentence extractor

We used the features from Munteanu's paper that are considered to be helpful in pinpointing parallel pairs:

- lengths of the sentences, as well as the length diference and length ratio (LENGTH);
- number and percentage of connected words for both F → E and E → F (TRANS);
- top three fertilities and their percentage in both F and E (FERT);
- length and percentage of the longest substrings which are not connected (UNCONNECT);
- length and percentage of longest word span which is connected (CONTIG).

Once we get the real-value of feature vectors, the ME principle is applied as equation 1 and a log-linear combination function is parameterized with positive and negative samples.

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

The extraction process

Our framework aims to find parallel sentence pairs from document aligned comparable corpus. After all sentence pairs from the target language document and the source language document are generated, we apply simple heuristic rules to filter the candidate pairs. Firstly, we verify that the ratio of the lengths of the two sentences is no greater than 2. Secondly, we ensure that at least half of the words in each sentence have a translation in the other sentence, according to the GIZA lexicon. Pairs that cannot satisfy these two criteria are filtered out.

In the next step, the ME model is used as a classifier. For each sentence pair, the features are extracted and converted into a 50 dimensional vector. Equation 1 is used (λ_j are fixed) so that the class label c_i for which we obtain the largest $P(c_i|sp)$ will be considered as the classification result.

Our implementation of ME training and classification is based on a freely available maximum entropy classifier written in C++ which can be downloaded from the following URL: http://www-tsujii.is.s.u-tokyo.ac.jp/tsuruoka/maxent/.

2.3.1. Experiments and Results

We choose German and English as source and target languages respectively for evaluation. In general, our experiments investigate two aspects: the accuracy of ME classifier (for internal development) and the improvement on an SMT system as a final evaluation.

Feature setting

The initial corpus is obtained by merging Europarl-v6 (Koehn, 2005) and News-Commentary 2010 corpora[9] (NC10). 11k sentences pairs are randomly selected from this corpus as positive examples; as we mentioned before, simply scrambling the order of sentence appearance in these positive samples, we generate 11k pairs as negative examples. Furthermore, 10k pairs are split as the training set and the rest 1k pairs are the test set.

We empirically investigate how the feature setting impact the result of classification with gradually adding features. Table 13 shows the performance under different features.

Table 13: ME performance on the development set

Feature sets	Precision	Recall	F-measure
LENGTH TRANS	0.794	0.786	0.790
+FERT	0.801	0.793	0.797
+UNCONNECT	0.817	0.804	0.810
+CONTIG	0.831	0.821	0.826

As we can see, basic features such as translation number and sentence length ratio are useful to distinguish between parallel and non-parallel sentences. In comparison with other additional features, the longest continuous span is important and significantly boosts the performance while the top 3 fertilities are not as helpful as we would expect.

[9] http://www.statmt.org/wmt10/training-parallel.tgz

Evaluation on an MT system

To evaluate the correctness of extracted sentence pairs, we add them to parallel corpora used for training translation models for a new MT system. We use Moses-EMS (Koehn et al., 2007) as the de facto standard for SMT systems and Multi-BLEU score to measure the improvement when adding the new sentence pairs. Table 14 gives an explanation on the various types of data. As it shows, although a large amount of sentence pairs are generated as candidates, the remaining (final) parallel sentence pairs are a small percentage of the possible candidates.

Table 14: Initial corpora size and the ratio of extracted parallel sentences

Document no.	Candidate sentences no.	Extracted parallel sentences no.
556,499	80,595,885	71,571

Table 3 indicates the BLEU score on the test corpus 'Balanced'. As the average length of extracted parallel sentences is 165, we train the system with different maximum lengths of 80 and 100. Moreover, because the German lexicon contains compound words, we investigate the configuration 'compound splitter' in the source language.

Table 15: SMT evaluation with different settings

Base corpus	Corpus size Sentence pair no.	Baseline (BLEU score)	With ME-extracted parallel sentences (BLEU score)	With Maximum length and compound split (BLEU score)
NC10	100,269	19.45	20.21	21.21
NC10+Europarl-v6	1,875,419	28.31	28.40	28.62

From this table we conclude that parallel sentence extraction can contribute to a better MT system but due to the scarcity of parallel data, the improvement of the best DE-EN MT system is not significant. How to extract more parallel data from a comparable corpus will be the focus of our future work. In addition, we postulate that the Wikipedia corpus cannot be helpful for the news domain. Exploring the improvement of MT when adding parallel data extracted from a comparable corpus with the same domain is to be investigated as well.

3. Conclusions

We have developed and tested a multitude of algorithms that address problems such as document alignment and parallel data mining, problems which are the focus of this deliverable. Some algorithms were designed specifically with these problems in mind: EMACC, an EM document aligner, PEXACC, a parallel phrase extractor from comparable corpora and our implementation of Munteanu's method of parallel sentence extraction from comparable corpora. Other algorithms (originally developed for a different problem) could also serve to solve some of the problems of interest. Thus, the comparability metric algorithm from section 1.4 and the document pair classifier from section 1.3 were designed to label a pair of documents as to their parallelism degree: "parallel", "strongly comparable", "weakly comparable" or "not comparable" but these labels could also be regarded as a measure of the document (translation) similarity. Thus, these could serve as indicating which source documents map to which target documents at a certain parallelism degree (in subsequent processing, an application as PEXACC may choose what type of pairs to process).

Having more than one method of solving the document alignment problem/parallel textual unit extraction problem does not mean that the work was duplicated. With respect to the document alignment problem, we now have different insights on what it means for two documents to be comparable and on what level. These different perspectives will enable us to search for a more precise definition of comparability and in turn, to see to what extent parallel data extracted from these documents is able to improve MT. As to the parallel textual unit extraction problem, different methods are likely to complement each other and thus be able to furnish the complete solution when presented with a comparable corpus.

We saw that, in the absence of translation information, we cannot (yet) give a precise measure of what it means for two text fragments to be parallel. It is very important to develop "parallelism" measures that do not depend (that much) on the computational bilingual resources used (such as translation lexicons) such that, all pairs of text fragments extracted from a comparable corpus with a "parallelism" score over a certain threshold to be automatically considered parallel (and used as such in SMT training for instance) with the highest degree of confidence.

Last but not least, drawing from our preliminary success in improving SMT by using parallel data extracted from comparable corpora (see Table 15), it is very important to determine the amount of comparable corpora that will need to be collected in order to significantly improve MT. In other words, we need to establish the parallel data to non-parallel data ratio in different types of comparable corpora in order to be able to compute the comparable corpora size requirements.

4. References

Abdul-Rauf, S. Schwenk, H. 2009. Exploiting Comparable Corpora with TER and TERp. In Proceedings of the 2nd Workshop on Building and Using Comparable Corpora, ACL-IJCNLP 2009, pages 46–54, Suntec, Singapore, 6 August 2009. © 2009 ACL and AFNLP

Borman, S. 2009. The Expectation Maximization Algorithm. A short tutorial. Online at: http://www.isi.edu/natural-language/teaching/cs562/2009/readings/B06.pdf

Brown, P. F., Lai, J. C., and Mercer, R. L. 1991. Aligning sentences in parallel corpora. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, pp. 169–176, June 8-21, 1991, University of California, Berkeley, California, USA.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, 19(2): 263—311.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O.F. 2010. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pp. 17—53, Uppsala, Sweden, July 2010. © Association for Computational Linguistics.

Ceauşu, A. 2009. Statistical Machine Translation for Romanian. PhD Thesis, Romanian Academy (in Romanian).

Chen, S. F. 1993. Aligning Sentences in Bilingual Corpora Using Lexical Information. In Proceedings of the 31st Annual Meeting on Association for Computational Linguistics, pp. 9—16, Columbus, Ohio, USA.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 39(B):1—38.

Fung, P.,and Cheung, P. 2004. Mining Very-Non-Parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM. In Proceedings of EMNLP 2004, Barcelona, Spain: July 2004.

Gao, Q., and Vogel, S. 2008. Parallel implementations of word alignment tool. In Proceedings of ACL-08 HLT: Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49—57, June 20, 2008, The Ohio State University, Columbus, Ohio, USA.

Koehn, P., Och, F.J., and Marcu, D. 2003. Statistical Phrase-Based Translation. In Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL), May 27-June 1, Edmonton, Canada.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. Machine Translation Summit X, pp. 79--86, 2005.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. 2007. Moses: open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp 177--180. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics.

Melamed, I. D. 2001. Empirical methods for exploiting parallel texts. MIT Press, 198 pages, January 2001, ISBN-13: 978-0262133807.

Montalvo, S., Martinez, R., Casillas, A., and Fresno, V. 2006. Multilingual Document Clustering: a Heuristic Approach Based on Cognate Named Entities. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 1145–1152, Sydney, July 2006. © 2006 Association for Computational Linguistics.

Munteanu, D. S., and Marcu, D. 2002. Processing comparable corpora with bilingual suffix trees. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 289–295, July 6-7, 2002, University of Pennsylvania, Philadelphia, USA.

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 44th Annual Meeting of the ACL, pages 81–88, Sydney, July 2006. ©2006 Association for Computational Linguistics

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

Ogilvie, P., and Callan, J. 2001. Experiments using the Lemur toolkit. In Proceedings of *TREC 2001*, pp 103–108.

Quirk, C., Udupa, R., and Menezes, A. 2007. Generative Models of Noisy Translations with Applications to Parallel Fragment Extraction. In *Proceedings of the MT Summit XI*, pages 321–327, Copenhagen, Demark, September, 2007.

Tao, T., and Zhai, CX. 2002. *Mining Comparable Bilingual Text Corpora for Cross Language Information Integration*. In Proceedings of KDD'05, August 21-24, 2005, Chicago, Illinois, USA. Copyright 2005 ACM 1-59593-135-X/05/0008

Tufiş, D., Ion, R., Bozianu, L., Ceauşu, A., and Ştefănescu, D. 2008. Romanian Wordnet: Current State, New Applications and Prospects. In Attila Tanacs, Dora Csendes, Veronika Vincze, Christiane Fellbaum, and Piek Vossen (eds.), *Proceedings of 4th Global WordNet Conference, GWC-2008*, pp. 441-452, Szeged, Hungary, January 2008. University of Szeged, Hungary. ISBN-13: 978-963-482-854-9.

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

Annexes

Annex 1: Detailed Results of the EMACC Algorithm on Document Alignment

The left column represents the cutting threshold of the dictionary translation equivalents that are considered for document alignment. The right columns show the cutting thresholds on which the probabilities of translation equivalents are updated during the EM process. We are going to present scores for each combination of these two parameters and, in addition to that, for each combination of thresholds parameters, we measure precision and recall if the algorithm outputs only the top 30% / 70% of the document alignments it found or accuracy (P = R = Acc.) if the algorithm outputs all (100%) the alignments it found. Every cell in whatever table below presents the precision (top) and the recall (bottom) of the algorithm used to align the documents.

For each pair of languages, the first three tables report the performance figure of the EM algorithm with the D2 initial distribution on each type of corpora and the fourth table presents the alignment (also for each type of corpora) constructed from the D2 distribution only. One should compare the first three tables with the fourth noting that we:

- bolded the figures which are the highest per table (both a precision and a recall);
- grayed the background of the best figures which are higher between any of the first three tables and the corresponding column in the fourth one. Thus we have a visual measure of the improvement the EM brings to the simple document alignment with the D2 distribution.

Table 16: Document alignment: Slovene-English results of EMACC with D2 on parallel corpora

SL-EN (p, 532 docs.)		0.001	0.4	0.8
0.001	**0.3**	**0.98742** 0.29511	**0.98742** 0.29511	**0.98742** 0.29511
	0.7	0.95698 0.66917	0.95698 0.66917	0.95967 0.67105
	1	0.89097 **0.89097**	0.88345 0.88345	0.87218 0.87218
0.4	**0.3**	0.97484 0.29135	0.98113 0.29323	**0.98742** 0.29511
	0.7	0.94623 0.66165	0.96505 0.67481	0.96774 0.67669
	1	0.84962 0.84962	0.87593 0.87593	0.87030 0.87030
0.8	**0.3**	0.89308 0.26691	0.91194 0.27255	0.89937 0.26879
	0.7	0.73387 0.51315	0.71505 0.5	0.70161 0.49060

SL-EN (p, 532 docs.)		0.001	0.4	0.8
	1	0.56390 0.56390	0.54887 0.54887	0.54699 0.54699

Table 17: Document alignment: Slovene-English results of EMACC with D2 on strongly comparable corpora

SL-EN (cs, 302 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.78888 0.23509	0.85555 0.25496	0.85555 0.25496
	0.7	0.72511 0.50662	0.81042 0.56622	0.82464 0.57615
	1	0.67549 0.67549	0.76158 0.76158	0.76821 0.76821
0.4	**0.3**	0.93333 0.27814	**0.96666** 0.28807	0.93333 0.27814
	0.7	0.85308 0.59602	0.91943 0.64238	0.89573 0.62582
	1	0.75827 0.75827	0.83112 **0.83112**	0.80463 0.80463
0.8	**0.3**	0.76666 0.22847	0.71111 0.21192	0.74444 0.22185
	0.7	0.54028 0.37748	0.52606 0.36754	0.53554 0.37417
	1	0.39735 0.39735	0.37417 0.37417	0.38741 0.38741

Table 18: Document alignment: Slovene-English results of EMACC with D2 on weakly comparable corpora

SL-EN (cw, 961 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.51041 0.15296	0.54166 0.16233	0.57291 0.17169
	0.7	0.36607 0.25598	0.41517 0.29032	0.41815 0.29240
	1	0.27887 0.27887	0.323621 0.323621	0.31945 0.31945
0.4	**0.3**	0.71875 0.21540	**0.73958** 0.22164	0.73611 0.22060

SL-EN (cw, 961 docs.)		0.001	0.4	0.8
	0.7	0.51339 0.35900	0.55505 0.38813	0.53422 0.37356
	1	0.40582 0.40582	0.42767 **0.42767**	0.40998 0.40998
0.8	**0.3**	0.51041 0.15296	0.52430 0.15712	0.57986 0.17377
	0.7	0.36904 0.25806	0.35714 0.24973	0.36755 0.25702
	1	0.27263 0.27263	0.27575 0.27575	0.28511 0.28511

Table 19: Document alignment: Slovene-English baseline

SL-EN (D2)		p, 532d	cs, 302d	cw, 961d
0.001	**0.3**	0.96855 0.29001	**0.97777** 0.29139	0.56097 0.16788
	0.7	**0.98382** 0.68738	0.88151 0.61589	0.41877 0.29301
	1	0.93785 **0.93785**	0.77814 0.77814	0.34202 0.34202
0.4	**0.3**	0.97484 0.29245	0.9 0.26821	**0.65505** 0.19624
	0.7	0.96765 0.67735	0.86255 0.60264	0.49850 0.34864
	1	0.88301 0.88301	0.81456 **0.81456**	0.39874 **0.39874**
0.8	**0.3**	0.84713 0.25285	0.59770 0.17931	0.40209 0.12054
	0.7	0.67934 0.47528	0.50246 0.35172	0.29385 0.20545
	1	0.53992 0.53992	0.37241 0.37241	0.23060 0.23060

Table 20: Document alignment: Estonian-English results of EMACC with D2 on parallel corpora

ET-EN (p, 182 docs.)		0.001	0.4	0.8
0.001	0.3	**1** 0.29670	**1** 0.29670	**1** 0.29670
	0.7	0.98425 0.68681	0.99212 0.69230	0.98425 0.68681
	1	0.95054 0.95054	0.96153 **0.96153**	0.95054 0.95054
0.4	0.3	**1** 0.29670	**1** 0.29670	**1** 0.29670
	0.7	**1** 0.69780	**1** 0.69780	**1** 0.69780
	1	0.92857 0.92857	0.95604 0.95604	0.95604 0.95604
0.8	0.3	**1** 0.29670	0.98148 0.29120	0.98148 0.29120
	0.7	0.91338 0.63736	0.88188 0.61538	0.87401 0.60989
	1	0.76373 0.76373	0.75274 0.75274	0.74175 0.74175

Table 21: Document alignment: Estonian-English results of EMACC with D2 on strongly comparable corpora

ET-EN (cs, 987 docs.)		0.001	0.4	0.8
0.001	0.3	0.79322 0.23780	0.81123 0.22541	0.87324 0.21345
	0.7	0.61627 0.43089	0.69180 0.44	0.68761 0.41414
	1	0.52331 0.52331	0.51239 0.51239	0.50111 0.50111
0.4	0.3	0.82372 0.24695	0.86440 0.25914	**0.88135** 0.26422
	0.7	0.65697 0.45934	0.71075 0.49695	0.71220 0.49796
	1	0.51524 0.51524	0.55182 **0.55182**	0.54674 0.54674

ET-EN (cs, 987 docs.)		0.001	0.4	0.8
0.8	**0.3**	0.73220 0.21951	0.72881 0.21849	0.74237 0.22256
	0.7	0.50436 0.35264	0.52616 0.36788	0.52180 0.36483
	1	0.37195 0.37195	0.38719 0.38719	0.38414 0.38414

Table 22: Document alignment: Estonian-English results of EMACC with D2 on weakly comparable corpora

ET-EN (cw, 483 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.5 0.14937	0.52083 0.15560	0.52083 0.15560
	0.7	0.31157 0.21784	0.31750 0.22199	0.32047 0.22406
	1	0.23029 0.23029	0.23029 0.23029	0.24481 0.24481
0.4	**0.3**	0.53472 0.15975	0.53472 0.15975	**0.59027** 0.17634
	0.7	0.30860 0.21576	0.32937 0.23029	0.36498 0.25518
	1	0.23236 0.23236	0.24896 0.24896	0.27800 **0.27800**
0.8	**0.3**	0.375 0.11203	0.38888 0.11618	0.35416 0.10580
	0.7	0.21661 0.15145	0.21958 0.15352	0.19287 0.13485
	1	0.15975 0.15975	0.15975 0.15975	0.14107 0.14107

Table 23: Document alignment: Estonian-English baseline

ET-EN (D2)		p, 182d	cs, 987d	cw, 483d
0.001	**0.3**	**1** 0.29670	0.85034 0.25432	**0.48611** 0.14522
	0.7	0.99212 0.69230	0.70784 0.49542	0.28486 0.19917

ET-EN (D2)		p, 182d	cs, 987d	cw, 483d
	1	0.97802 **0.97802**	0.55137 0.55137	0.20954 0.20954
0.4	**0.3**	**1** 0.29670	**0.87030** 0.26100	0.46853 0.13987
	0.7	**1** 0.69780	0.73060 0.51074	0.33432 0.23382
	1	0.95054 0.95054	0.57727 **0.57727**	0.25678 **0.25678**
0.8	**0.3**	0.87037 0.26111	0.61111 0.18276	0.22377 0.06694
	0.7	0.824 0.57222	0.45845 0.32087	0.17065 0.11924
	1	0.72777 0.72777	0.34475 0.34475	0.13598 0.13598

Table 24: Document alignment: Romanian-English results of EMACC with D2 on parallel corpora

RO-EN (p, 21 docs.)		0.001	0.4	0.8
0.001	**0.3**	**1** 0.28571	**1** 0.28571	**1** 0.28571
	0.7	**1** 0.66666	**1** 0.66666	**1** 0.66666
	1	0.90476 0.90476	0.90476 0.90476	0.90476 0.90476
0.4	**0.3**	**1** 0.28571	**1** 0.28571	**1** 0.28571
	0.7	**1** 0.66666	**1** 0.66666	**1** 0.66666
	1	**1** **1**	0.90476 0.90476	0.90476 0.90476
0.8	**0.3**	**1** 0.28571	**1** 0.28571	**1** 0.28571
	0.7	0.85714 0.57142	0.85714 0.57142	0.92857 0.61904
	1	0.80952	0.80952	0.90476

		0.80952	0.80952	0.90476

Table 25: Document alignment: Romanian-English results of EMACC with D2 on strongly comparable corpora

RO-EN (cs, 42 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.58333 0.16666	0.75 0.21428	0.83333 0.23809
	0.7	0.58620 0.40476	0.65517 0.45238	0.72413 0.5
	1	0.47619 0.47619	0.54761 0.54761	0.61904 0.61904
0.4	**0.3**	**1** 0.28571	**1** 0.28571	**1** 0.28571
	0.7	0.89655 0.61904	**1** 0.69047	**1** 0.69047
	1	0.76190 0.76190	0.85714 **0.85714**	0.85714 **0.85714**
0.8	**0.3**	**1** 0.28571	**1** 0.28571	**1** 0.28571
	0.7	0.82758 0.57142	0.86206 0.59523	0.86206 0.59523
	1	0.73809 0.73809	0.73809 0.73809	0.69047 0.69047

Table 26: Document alignment: Romanian-English results of EMACC with D2 on weakly comparable corpora

RO-EN (cw, 68 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.65 0.19117	0.65 0.19117	0.65 0.19117
	0.7	0.46808 0.32352	0.51063 0.35294	0.48936 0.33823
	1	0.36764 0.36764	0.38235 0.38235	0.36764 0.36764
0.4	**0.3**	**1** 0.29411	0.9 0.26470	0.9 0.26470
	0.7	0.87234 0.60294	0.85106 0.58823	0.85106 0.58823

RO-EN (cw, 68 docs.)		0.001	0.4	0.8
	1	0.66176 **0.66176**	0.64705 0.64705	0.64705 0.64705
0.8	**0.3**	0.95 0.27941	0.85 0.25	0.85 0.25
	0.7	0.68085 0.47058	0.65957 0.45588	0.65957 0.45588
	1	0.51470 0.51470	0.47058 0.47058	0.48529 0.48529

Table 27: Document alignment: Romanian-English baseline

RO-EN (D2)		p, 21d	cs, 42d	cw, 68d
0.001	**0.3**	**1** 0.28571	0.83333 0.23809	0.65 0.19117
	0.7	**1** 0.66666	0.86206 0.59523	0.48936 0.33823
	1	0.90476 0.90476	0.71428 0.71428	0.35294 0.35294
0.4	**0.3**	**1** 0.28571	**1** 0.28571	**0.85** 0.25
	0.7	**1** 0.66666	**1** 0.69047	0.78723 0.54411
	1	0.90476 0.90476	0.85714 **0.85714**	0.61764 **0.61764**
0.8	**0.3**	**1** 0.28571	**1** 0.29268	0.73684 0.21212
	0.7	**1** 0.66666	0.75 0.51219	0.52173 0.36363
	1	**1** **1**	0.56097 0.56097	0.37878 0.37878

Table 28: Document alignment: Greek-English results on parallel corpora

EL-EN (p, 87 docs.)		0.001	0.4	0.8
0.001	**0.3**	**1** 0.29885	**1** 0.29885	**1** 0.29885

EL-EN (p, 87 docs.)		0.001	0.4	0.8
	0.7	**1** 0.68965	**1** 0.68965	**1** 0.68965
	1	**1** **1**	**1** **1**	**1** **1**
0.4	**0.3**	**1** 0.29885	**1** 0.29885	**1** 0.29885
	0.7	**1** 0.68965	**1** 0.68965	**1** 0.68965
	1	**1** **1**	**1** **1**	**1** **1**
0.8	**0.3**	0.96153 0.28735	**1** 0.29885	**1** 0.29885
	0.7	0.7 0.48275	0.65 0.44827	0.63333 0.43678
	1	0.52873 0.52873	0.49425 0.49425	0.51724 0.51724

Table 29: Document alignment: Greek-English results of EMACC with D2 on strongly comparable corpora

EL-EN (cs, 407 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.95081 0.28501	0.96721 0.28992	**0.97540** 0.29238
	0.7	0.83450 0.58230	0.94014 0.65601	0.93661 0.65356
	1	0.70761 0.70761	0.80098 **0.80098**	0.77641 0.77641
0.4	**0.3**	0.92622 0.27764	0.95081 0.28501	0.95081 0.28501
	0.7	0.77464 0.54054	0.81338 0.56756	0.82394 0.57493
	1	0.62899 0.62899	0.62653 0.62653	0.63882 0.63882
0.8	**0.3**	0.54098 0.16216	0.54918 0.16461	0.54098 0.16216
	0.7	0.33098 0.23095	0.31690 0.22113	0.30281 0.21130

	1	0.23832 0.23832	0.23095 0.23095	0.22358 0.22358

Table 30: Document alignment: Greek-English results of EMACC with D2 on weakly comparable corpora

EL-EN (cw, 352 docs.)		**0.001**	**0.4**	**0.8**
0.001	**0.3**	0.10476 0.03125	0.11428 0.03409	**0.15238** 0.04545
	0.7	0.07317 0.05113	0.09349 0.06534	0.10569 0.07386
	1	0.05681 0.05681	0.06534 0.06534	0.07670 **0.07670**
0.4	**0.3**	0.10476 0.03125	0.12380 0.03693	0.10476 0.03125
	0.7	0.07723 0.05397	0.08536 0.05965	0.06504 0.04545
	1	0.0625 0.0625	0.0625 0.0625	0.05113 0.05113
0.8	**0.3**	0.07619 0.02272	0.06666 0.01988	0.06666 0.01988
	0.7	0.03658 0.02556	0.03252 0.02272	0.03658 0.02556
	1	0.02840 0.02840	0.02272 0.02272	0.02556 0.02556

Table 31: Document alignment: Greek-English baseline

EL-EN (D2)		**p, 87d**	**cs, 407d**	**cw, 352d**
0.001	**0.3**	**1** 0.29885	**0.94214** 0.28148	0.06666 0.01988
	0.7	**1** 0.68965	0.87985 0.61481	0.06097 0.04261
	1	**1** **1**	0.71851 **0.71851**	0.04829 0.04829
0.4	**0.3**	**1** 0.29411	0.88429 0.26419	**0.11428** 0.03428
	0.7	**1** 0.69411	0.68197 0.47654	0.07786 0.05428

	1	0.95294 0.95294	0.54567 0.54567	0.06285 **0.06285**
0.8	**0.3**	**1** 0.29761	0.54782 0.16406	0.03883 0.01162
	0.7	0.89655 0.61904	0.36567 0.25520	0.03333 0.02325
	1	0.72619 0.72619	0.27083 0.27083	0.02325 0.02325

Table 32: Document alignment: Lithuanian-English results of EMACC with D2 on parallel corpora

LT-EN (p, 347 docs.)		**0.001**	**0.4**	**0.8**
0.001	**0.3**	0.99038 0.29682	0.98076 0.29394	0.99038 0.29682
	0.7	0.94628 0.65994	0.93801 0.65417	0.96694 0.67435
	1	0.92795 0.92795	0.91354 0.91354	0.93371 **0.93371**
0.4	**0.3**	**1** 0.29971	0.99038 0.29682	0.99038 0.29682
	0.7	0.94214 0.65706	0.95041 0.66282	0.95867 0.66858
	1	0.88184 0.88184	0.91066 0.91066	0.91066 0.91066
0.8	**0.3**	0.95192 0.28530	0.95192 0.28530	0.95192 0.28530
	0.7	0.90082 0.62824	0.90495 0.63112	0.90495 0.63112
	1	0.79538 0.79538	0.83861 0.83861	0.81844 0.81844

Table 33: Document alignment: Lithuanian-English results of EMACC with D2 on strongly comparable corpora

LT-EN (cs, 507 docs.)		**0.001**	**0.4**	**0.8**
0.001	**0.3**	0.875 0.26232	0.92105 0.27613	0.92105 0.27613

LT-EN (cs, 507 docs.)		0.001	0.4	0.8
	0.7	0.79661 0.55621	0.83615 0.58382	0.84180 0.58777
	1	0.65088 0.65088	0.70808 0.70808	0.69625 0.69625
0.4	**0.3**	0.96710 0.28994	0.96710 0.28994	**0.97368** 0.29191
	0.7	0.84745 0.59171	0.86440 0.60355	0.87005 0.60749
	1	0.69230 0.69230	0.72978 **0.72978**	0.72583 0.72583
0.8	**0.3**	0.91447 0.27416	0.92105 0.27613	0.90789 0.27218
	0.7	0.69491 0.48520	0.71186 0.49704	0.70338 0.49112
	1	0.52071 0.52071	0.52859 0.52859	0.52071 0.52071

Table 34: Document alignment: Lithuanian-English results of EMACC with D2 on weakly comparable corpora

LT-EN (cw, 325 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.44329 0.13230	0.50515 0.15076	0.47422 0.14153
	0.7	0.26872 0.18769	0.28193 0.19692	0.27312 0.19076
	1	0.2 0.2	0.20307 0.20307	0.20923 0.20923
0.4	**0.3**	0.51546 0.15384	0.54639 0.16307	**0.55670** 0.16615
	0.7	0.35682 0.24923	0.35242 0.24615	0.38766 0.27076
	1	0.26461 0.26461	0.25846 0.25846	0.28307 **0.28307**
0.8	**0.3**	0.36082 0.10769	0.37113 0.11076	0.38144 0.11384
	0.7	0.22026 0.15384	0.22026 0.15384	0.21145 0.14769

	1	0.15692 0.15692	0.16307 0.16307	0.15384 0.15384

Table 35: Document alignment: Lithuanian-English baseline

LT-EN (D2)		**p, 347d**	**cs, 507d**	**cw, 325d**
0.001	**0.3**	0.88461 0.26512	**0.95364** 0.28514	0.50515 0.15076
	0.7	0.90495 0.63112	0.86402 0.60396	0.26431 0.18461
	1	0.90778 **0.90778**	0.72673 **0.72673**	0.19076 0.19076
0.4	**0.3**	**0.95192** 0.28530	0.88741 0.26587	**0.60416** 0.18012
	0.7	0.89669 0.62536	0.81818 0.57142	0.33777 0.23602
	1	0.89913 0.89913	0.70039 0.70039	0.24844 **0.24844**
0.8	**0.3**	0.94117 0.28152	0.74324 0.22267	0.35416 0.10625
	0.7	0.82773 0.57771	0.57101 0.39878	0.20535 0.14375
	1	0.75659 0.75659	0.47975 0.47975	0.15937 0.15937

Table 36: Document alignment: Latvian-English results of EMACC with D2 on parallel corpora

LV-EN (p, 184 docs.)		**0.001**	**0.4**	**0.8**
0.001	**0.3**	**1** 0.29891	**1** 0.29891	**1** 0.29891
	0.7	**1** 0.69565	**1** 0.69565	**1** 0.69565
	1	0.98913 0.98913	0.98913 0.98913	0.98913 0.98913
0.4	**0.3**	**1** 0.29891	**1** 0.29891	**1** 0.29891

LV-EN (p, 184 docs.)		0.001	0.4	0.8
	0.7	**1** 0.69565	**1** 0.69565	**1** 0.69565
	1	**1** **1**	**1** **1**	0.98913 0.98913
0.8	**0.3**	**1** 0.29891	**1** 0.29891	**1** 0.29891
	0.7	0.99218 0.69021	0.97656 0.67934	0.97656 0.67934
	1	0.89130 0.89130	0.89130 0.89130	0.89673 0.89673

Table 37: Document alignment: Latvian-English results of EMACC with D2 on strongly comparable corpora

LV-EN (cs, 560 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.88484 0.26497	0.90909 0.27223	0.91515 0.27404
	0.7	0.85194 0.59528	0.89870 0.62794	0.90389 0.63157
	1	0.73684 0.73684	0.79310 0.79310	0.79854 **0.79854**
0.4	**0.3**	0.95151 0.28493	**0.95757** 0.28675	**0.95757** 0.28675
	0.7	0.87272 0.60980	0.90909 0.63520	0.90129 0.62976
	1	0.74591 0.74591	0.78584 0.78584	0.77676 0.77676
0.8	**0.3**	0.89696 0.26860	0.92727 0.27767	0.93939 0.28130
	0.7	0.70909 0.49546	0.73766 0.51542	0.72987 0.50998
	1	0.56261 0.56261	0.58076 0.58076	0.57713 0.57713

Table 38: Document alignment: Latvian-English results of EMACC with D2 on weakly comparable corpora

LV-EN (cw, 511 docs.)	0.001	0.4	0.8

LV-EN (cw, 511 docs.)		0.001	0.4	0.8
0.001	**0.3**	0.20915 0.06262	0.17647 0.05283	0.21568 0.06457
	0.7	0.10364 0.07240	0.10084 0.07045	0.10924 0.07632
	1	0.07436 0.07436	0.07240 0.07240	0.07827 0.07827
0.4	**0.3**	0.20261 0.06066	**0.23529** 0.07045	**0.23529** 0.07045
	0.7	0.13445 0.09393	0.14285 0.09980	0.13725 0.09589
	1	0.09589 0.09589	0.10176 **0.10176**	0.09589 0.09589
0.8	**0.3**	0.12418 0.03718	0.13725 0.04109	0.14379 0.04305
	0.7	0.06442 0.04500	0.07002 0.04892	0.07282 0.05088
	1	0.05870 0.05870	0.05479 0.05479	0.05870 0.05870

Table 39: Document alignment: Latvian-English baseline

LV-EN (D2)		p, 184d	cs, 560d	cw, 511d
0.001	**0.3**	**1** 0.29891	**0.91463** 0.27322	0.12418 0.03718
	0.7	0.99218 0.69021	0.91406 0.63934	0.07563 0.05283
	1	0.97826 **0.97826**	0.80692 **0.80692**	0.05870 0.05870
0.4	**0.3**	**1** 0.29891	0.90853 0.27239	**0.13071** 0.03921
	0.7	0.98437 0.68478	0.90837 0.63436	0.10924 0.07647
	1	0.97826 **0.97826**	0.80621 0.80621	0.09803 **0.09803**
0.8	**0.3**	0.96296 0.28415	0.79878 0.23948	0.07947 0.02380

LV-EN (D2)		p, 184d	cs, 560d	cw, 511d
	0.7	0.96875 0.67759	0.63350 0.44241	0.05681 0.03968
	1	0.89071 0.89071	0.53382 0.53382	0.04761 0.04761

Annex 2: Detailed Results of the EMACC Algorithm on Paragraph Alignment

The task consisted of aligning 200 (parallel) paragraphs (at most 50 words per paragraph) extracted from JRC Acquis Corpus (http://langtech.jrc.it/JRC-Acquis.html) for each pair of languages of the project using the EMACC algorithm with the same settings as in the case of document alignments.

The left column represents the cutting threshold of the dictionary translation equivalents that are considered for paragraph alignment. The right columns show the cutting thresholds on which the probabilities of translation equivalents are updated during the EM process. We are going to present scores for each combination of these two parameters and, in addition to that, for each combination of thresholds parameters, we measure precision and recall if the algorithm outputs only the top 30% / 70% of the paragraph alignments it found or accuracy (P = R = Acc.) if the algorithm outputs all (100%) the alignments it found. Every cell in whatever table below presents the precision (top) and the recall (bottom) of the algorithm used to align the documents.

For each pair of languages, the first table reports the performance figure of the EM algorithm with the D2 initial distribution on parallel collection of paragraphs and the second table presents the alignment constructed from the D2 distribution only. One should compare the tables noting that we:

- bolded the figures which are the highest per table (both a precision and a recall);
- grayed the background of the best figures which are higher between the first table and the corresponding column in the second one. Thus we have a visual measure of the improvement the EM brings to the simple document alignment with the D2 distribution.

Table 40: Paragraph alignment: Slovene-English results of EMACC with D2

SL-EN (p, 200 pars.)		**0.001**	**0.4**	**0.8**
0.001	**0.3**	0.96666 0.29	0.96666 0.29	0.98333 0.295
	0.7	0.92857 0.65	0.92142 0.645	0.95 0.665
	1	0.795 0.795	0.83 **0.83**	0.815 0.815
0.4	**0.3**	**1** 0.3	**1** 0.3	**1** 0.3
	0.7	0.94285 0.66	0.95 0.665	0.96428 0.675
	1	0.775 0.775	0.78 0.78	0.775 0.775
0.8	**0.3**	0.75 0.225	0.78333 0.235	0.75 0.225

SL-EN (p, 200 pars.)		0.001	0.4	0.8
	0.7	0.55 0.385	0.51428 0.36	0.50714 0.355
	1	0.41 0.41	0.395 0.395	0.385 0.385

Table 41: Slovene-English baseline

SL-EN (D2)		p, 200p
0.001	0.3	**0.94915** 0.28140
	0.7	0.92805 0.64824
	1	0.84924 **0.84924**
0.4	0.3	0.94827 0.28205
	0.7	0.86029 0.6
	1	0.74871 0.74871
0.8	0.3	0.625 0.18617
	0.7	0.51145 0.35638
	1	0.41489 0.41489

Table 42: Paragraph alignment: Estonian-English results of EMACC with D2

ET-EN (p, 200 pars.)		0.001	0.4	0.8
0.001	0.3	0.98333 0.295	**1** 0.3	**1** 0.3
	0.7	0.97142 0.68	0.95714 0.67	0.98571 0.69
	1	0.875 0.875	0.895 0.895	0.92 **0.92**
0.4	0.3	0.93333 0.28	0.93333 0.28	0.93333 0.28
	0.7	0.84285 0.59	0.79285 0.555	0.77857 0.545
	1	0.69 0.69	0.64 0.64	0.64 0.64
0.8	0.3	0.68333 0.205	0.63333 0.19	0.65 0.195
	0.7	0.47142 0.33	0.43571 0.305	0.44285 0.31
	1	0.345 0.345	0.32 0.32	0.32 0.32

Table 43: Paragraph alignment: Estonian-English baseline

ET-EN (D2)		p, 200p
0.001	0.3	**0.94915** 0.28282
	0.7	0.92028 0.64141
	1	0.86868 **0.86868**
0.4	0.3	0.81355 0.24365
	0.7	0.68613 0.47715
	1	0.60913 0.60913
0.8	0.3	0.43396

ET-EN (D2)		p, 200p
		0.12994
	0.7	0.37398 0.26136
	1	0.33333 0.33333

Table 44: Paragraph alignment: Romanian-English results of EMACC with D2

RO-EN (p, 200 pars.)		0.001	0.4	0.8
0.001	**0.3**	0.98333 0.295	0.98333 0.295	**1** 0.3
	0.7	0.93571 0.655	0.95 0.665	0.94285 0.66
	1	0.84 0.84	0.865 **0.865**	0.86 0.86
0.4	**0.3**	**1** 0.3	**1** 0.3	**1** 0.3
	0.7	0.94285 0.66	0.96428 0.675	0.95714 0.67
	1	0.83 0.83	0.855 0.855	0.86 0.86
0.8	**0.3**	0.91666 0.275	0.91666 0.275	0.9 0.27
	0.7	0.72857 0.51	0.69285 0.485	0.67142 0.47
	1	0.56 0.56	0.525 0.525	0.52 0.52

Table 45: Paragraph alignment: the Romanian-English baseline

RO-EN (D2)		p, 200p
0.001	**0.3**	0.96610 0.28643
	0.7	0.95683 0.66834
	1	0.87437

RO-EN (D2)		p, 200p
		0.87437
0.4	**0.3**	**1** 0.29797
	0.7	0.93478 0.65151
	1	0.85353 0.85353
0.8	**0.3**	0.68965 0.20408
	0.7	0.57352 0.4
	1	0.46938 0.46938

Table 46: Paragraph alignment: Greek-English results of EMACC with D2

EL-EN (p, 200 pars.)		0.001	0.4	0.8
0.001	**0.3**	**0.98333** 0.295	**0.98333** 0.295	**0.98333** 0.295
	0.7	0.92857 0.65	0.92857 0.65	0.93571 0.655
	1	0.79 0.79	0.82 0.82	0.83 **0.83**
0.4	**0.3**	0.78333 0.235	0.81666 0.245	0.81666 0.245
	0.7	0.57857 0.405	0.57857 0.405	0.57142 0.4
	1	0.465 0.465	0.46 0.46	0.46 0.46
0.8	**0.3**	0.61666 0.185	0.56666 0.17	0.55 0.165
	0.7	0.35714 0.25	0.33571 0.235	0.34285 0.24
	1	0.26 0.26	0.235 0.235	0.24 0.24

Table 47: Paragraph alignment: the Greek-English baseline

EL-EN (D2)		p, 200p
0.001	**0.3**	**0.94915** 0.28140
	0.7	0.89208 0.62311
	1	0.80904 **0.80904**
0.4	**0.3**	0.71929 0.21243
	0.7	0.62222 0.43523
	1	0.52331 0.52331
0.8	**0.3**	0.45454 0.13605
	0.7	0.36274 0.25170
	1	0.31292 0.31292

Table 48: Paragraph alignment: Lithuanian-English results of EMACC with D2

LT-EN (p, 200 pars.)		0.001	0.4	0.8
0.001	**0.3**	**0.96666** 0.29	**0.96666** 0.29	**0.96666** 0.29
	0.7	0.84285 0.59	0.88571 0.62	0.89285 0.625
	1	0.75 0.75	0.76 0.76	0.765 **0.765**
0.4	**0.3**	0.85 0.255	0.93333 0.28	0.91666 0.275
	0.7	0.71428 0.5	0.68571 0.48	0.68571 0.48
	1	0.595 0.595	0.56 0.56	0.555 0.555

LT-EN (p, 200 pars.)		0.001	0.4	0.8
0.8	0.3	0.65 0.195	0.6 0.18	0.55 0.165
	0.7	0.47857 0.335	0.47142 0.33	0.43571 0.305
	1	0.36 0.36	0.36 0.36	0.335 0.335

Table 49: Paragraph alignment: the Lithuanian-English baseline

LT-EN (D2)		p, 200p
0.001	0.3	**0.96610** 0.28643
	0.7	0.84892 0.59296
	1	0.72864 **0.72864**
0.4	0.3	0.74576 0.22110
	0.7	0.66187 0.46231
	1	0.56783 0.56783
0.8	0.3	0.44642 0.13297
	0.7	0.31297 0.21808
	1	0.27127 0.27127

Table 50: Paragraph alignment: Latvian-English results of EMACC with D2:

LV-EN (p, 200 pars.)		0.001	0.4	0.8
0.001	0.3	**1** 0.3	**1** 0.3	**1** 0.3
	0.7	0.97857 0.685	0.97142 0.68	0.97857 0.685
	1	0.88 **0.88**	0.875 0.875	0.875 0.875
0.4	0.3	**1** 0.3	0.98333 0.295	0.98333 0.295
	0.7	0.9 0.63	0.9 0.63	0.89285 0.625
	1	0.755 0.755	0.765 0.765	0.775 0.775
0.8	0.3	0.86666 0.26	0.83333 0.25	0.81666 0.245
	0.7	0.67142 0.47	0.62142 0.435	0.62142 0.435
	1	0.54 0.54	0.48 0.48	0.48 0.48

Table 51 Paragraph alignment: the Latvian-English baseline

LV-EN (D2)		p, 200p
0.001	0.3	**0.96610** 0.28643
	0.7	0.95683 0.668341
	1	0.85427 **0.85427**
0.4	0.3	0.89830 0.26767
	0.7	0.78985 0.55050
	1	0.70707 0.70707

LV-EN (D2)		p, 200p
0.8	**0.3**	0.625 0.18617
	0.7	0.46564 0.32446
	1	0.36702 0.36702

Table 52 Paragraph alignment: German-English results of EMACC with D2

DE-EN (p, 200 pars.)		0.001	0.4	0.8
0.001	**0.3**	0.78333 0.235	0.78333 0.235	0.81666 0.245
	0.7	0.78571 0.55	0.76428 0.535	0.80714 0.565
	1	0.735 0.735	0.685 0.685	0.74 **0.74**
0.4	**0.3**	0.86666 0.26	0.83333 0.25	**0.88333** 0.265
	0.7	0.74285 0.52	0.71428 0.5	0.75 0.525
	1	0.63 0.63	0.605 0.605	0.62 0.62
0.8	**0.3**	0.81666 0.245	0.83333 0.25	0.85 0.255
	0.7	0.56428 0.395	0.55714 0.39	0.55 0.385
	1	0.445 0.445	0.415 0.415	0.41 0.41

Table 53 Paragraph alignment: the German-English baseline

DE-EN (D2)		p, 200p
0.001	**0.3**	0.77966 0.23350
	0.7	0.78832 0.54822

DE-EN (D2)		p, 200p
	1	0.74619 **0.74619**
0.4	**0.3**	**0.89655** 0.26666
	0.7	0.76470 0.53333
	1	0.63589 0.63589
0.8	**0.3**	0.51851 0.15469
	0.7	0.5 0.34806
	1	0.40331 0.40331