STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 1, 2010

MODEL ALIGNMENT BY USING THE CONCEPT DEFINITIONS

ADELA SÎRBU[1,2], LAURA DIOŞAN[1,2], ALEXANDRINA ROGOZAN[1], JEAN-PIERRE PECUCHET[1]

Abstract. The alignment between two dictionaries will certainly improve the performances of the information retrieval process. We develop a custom terminology alignment by an SVM classifier with an optimised kernel trained on a compact, but relevant representation of such definition pairs by several similarity measures and the length of definitions. The aligner was trained on a database of aligned definitions that was semi-automatically created by using the Coma++ tool. The results obtained on the test set show the relevance of our approach.

1. Introduction

One of the goals of the ASICOM project[1] is to improve the fusion between a specialized terminology and a general vocabulary employed by a neophyte user in order to retrieve documents on Internet. These goals could be summarised as follows:

- to design a model alignment in order to help or to guide the automatically transformation of dictionaries;
- to quantify the role of ontologies and/or hierarchies of concepts/dictionaries for the model transformation;
- to align two concepts from their definitions only, or from their definitions and their paths in the hierarchies of concepts, or from their definitions, their paths and their fathers in the hierarchies of concepts.

The goal of the mapping between two hierarchies of concepts (dictionaries) is to align a concept from a dictionary with a concept of another dictionary

Received by the editors: January 10, 2009.

2010 *Mathematics Subject Classification.* 68T05,91E45.

1998 *CR Categories and Descriptors.* code I.2.6 [**Learning**]: – *Concept learning*.

Key words and phrases. Concept alignment, Machine Learning, Binary Classification, Support Vector Machine.

[1]Architecture de Systme d'information Interoprable pour les industries du Commerce, project financed by *Pôle de compétitivité Industries du Commerce* and *Logistique Seine-Normandie, 2009-2011*

by using their definitions, since their names may be different. In other words, the purpose is to identify concepts semantically identical, even though their concept labels are different.

The correspondence may be such that:

- a concept from a dictionary matches even several concepts from another dictionary. In other words, several concepts of the second dictionary can then be labelled with the same concept from the first dictionary. These concepts could then be seen as "the result of" the concept of the first dictionary. We actually deal by a "sort of" relationship.
- a concept from a dictionary matches at most one concept from another dictionary. In this case, the two concepts are so similar or equivalent; therefore we discuss about an "equivalence" relation.

Of course, another relationships could be discovered among the content of two dictionaries.

In order to perform some numerical experiments two dictionary were actually considered: ASICOM_CCL08A (or, shorter, CCL) and Customs_WCO (WCO). The first dictionary contains 2191 of concepts and for each concepts information about id, object class term (or father in the hierarchy), property term, representation term, entry name (in fact, label and path) and explanation are retained. In the case of WCO dictionary, we deal with 264 concepts for which we have retained the id, the entry name (WCO name or label and UNTDED Name or path in the hierarchy of concepts), data model class (actually, the concept father) and the explanation.

In order to perform our analysis, we have to build a database with couples of definition aligned. This base could be realised in a semi-automatically manner or manually, by a specialist area. Because our aim is to design a human-knowledge independent application, we have decided to use a database that was constructed in a semi-automatically manner by using a special tool: Coma++.

The definitions aligned by Coma++ are utilised as train data for a Machine Learning algorithm in order to aligned unseen definitions. In fact, the database constructed by Coma++ help us to enable the self-aligning process, as well as it serves as a repository for automatic alignment. The Coma++ principles could also be used as a complementary technique to discover alignments not seen by the automatic alignment.

This paper is structured as follows: Section 2 gives a short review of different alignment models. Section 3 details the characteristics of the corpora and of the linguistic treatments that have been performed. The alignment model is described and analysed (through several numerical experiments) in the next two sections. Finally, Section 6 concludes the paper.

2. Related work

To our knowledge, only the problem of aligning sentences from parallel bilingual corpora has been intensively studied for automated translation. While much of research has focused on the unsupervised models [1, 4], a number of supervised discriminatory approaches have been recently proposed for automatic alignment [2, 10, 12].

One of the first algorithms used to align parallel corpora proposed by Brown [1] is based solely on the number of words/characters in each sentence. Chen [4] has developed a simple statistical word-to-word translation model. Dynamic programming, at the level of words, performs the search of the best alignment in these models.

Related to the use of linguistic information a more recent work [11] shows the benefit of combining multilevel linguistic representations (these levels refer to morphological, syntactic and semantic analyses). Moreover, data fusion has been exhaustively investigated in the literature, especially in the framework of Information Retrieval [11].

Concerning the supervised methods, Taskar et al. [12] have cast the word alignment as a maximum weighted matching problem where each pair of sentences has associated a score function, which reflects the desirability of the alignment of that pair. The alignment for the sentence pair corresponds to the highest scoring matching under some constraints (for instance, the requirement that matching be one-to-one). Moore [10] has introduced a hybrid and supervised approach that adapts and combines the sentence-length-based methods with the word-correspondence-based methods. Ceausu [2] has proposed another supervised hybrid method that uses a Support Vector Machine (SVM) classifier [14] to distinguish between aligned and non-aligned examples of sentence pairs; each pair has been represented by a set of statistical characteristics (like translation equivalence, word sentence length correlation, character sentence length correlation, word rank correlation, non-word sentence length correlation).

The model we develop in what follows borrows some aspects from Moore and Ceausu's approaches, but it is enriched with several new elements. Our model considers the alignment task as a classification problem (as in Ceausu's case). Although, in our case, the information about definitions is organised based on several similarity measures, while the classification problem is solved by using an SVM algorithm, which involves an optimised kernel function.

3. The corpora and the linguistic processing

3.1. **Coma++.** Coma++ is actually a tool useful for semi-automatically alignment of concepts [8]. It is a schema and ontology matching tool. It

utilises a composite approach to combine different match algorithms. Furthermore, it offers a comprehensive infrastructure to solve large real-world matching problems. The graphical interface offers a variety of interactions, allowing the user to influence in the match process in many ways.

It is based on a composite schema matching and a flexible framework for combining matching algorithms. COMA++ supports a comprehensive and extensible library of individual matchers, which can be selected to perform a match operation. Using the GUI, it is easy to construct new, more powerful, matchers by combining existing ones. Moreover, it is possible to specify match strategies as work flows of multiple match steps, allowing to divide and successively solve complex match tasks in multiple stages [8].

Taken into account the alignments performed by Coma++ between definitions from CCL and WCO dictionaries, our aim was to develop a Machine Learning algorithm that will be able to put in correspondence more definitions of two different dictionaries and to improve the performance of alignment compared to Coma++. For this purpose a statistical learning based on SVM [14] is performed from a base of learning achieved after manual correction of alignments produced by Coma++.

The model we propose performs two important steps: represent, in a particular manner, the couples of definitions and than, learn or classify these couples.

Before we present our approach utilised to align the definitions, several details about the preliminary treatments of these definitions are provided.

3.2. **Linguistic processing.** As we already said, in order to automatically perform the alignment, several definitions are considered from two dictionaries: WCO and CCL. The English is the common language for both dictionaries. Each definition is retaining as a vector of words, each element of this vector being enriched only with its lemma and its synonyms.

The literature shows that a purely statistical approach on the plain text provides weak results for automatic text understanding. Several linguistic treatments, such as the labelling at the syntactic level (POS - Parts of speech - tagging) must be performed. Therefore, in order to achieve an efficient automatic classification "aligned" *vs.* "not aligned" of the definition couples, the following (structural and semantic) linguistic processing has been performed:

- segmentation – consists in cutting a sequence of characters so that various characters that form a single word can be brought together. Classic segmentation means cutting the sequences of characters depending on several separation characters such as "space", "tab" or "backspace";

- filtering of the stop words – *stop words* is the name given to words like *in*, *a*, *of*, *the*, *on* that are not representative should not be taken into consideration;
- bringing the words to the canonical form – in order to work directly with the words they must be brought to a canonical form. For instance the words *uses*, *using*, *used* refer to the same thing but under different forms, but if they are compared like this it will be obtained that they are different. There are different ways to solve this problem, and one of them could be to apply a stemming algorithm [13] – is the process of reducing inflected (or sometimes derived) words to their stem, base or root form. The stem does not have to be identical to the morphological root of the word; it is usually sufficient that related words map the same stem, even if this stem is not a valid root in itself.

3.3. **Similarity measures.** To enable a rapid and effective learning of definition alignment, we must avoid the problem associated with a classic representation based on the tf-idf[2] weighting scheme where the bags of words are translated into vectors of large sizes. In our case, the large size of such vectors is equal to the number of words contained by all the definitions chosen from all the dictionaries. In addition, the definitions could be considered as short text and thus, some sparse vectors will correspond to each definition. Therefore, we use several measures of similarity between two structures:

- the *Matching* coefficient [6] – it counts the common elements of the given structures.
- the *Dice* coefficient [7] – it is defined as twice the number of common elements, divided by the total number of elements,
- the *Jaccard* coefficient [9] – it is defined as the number of common elements, divided by the total number of elements,
- the *Overlap* coefficient [5] – it is defined as the number of common elements, divided by the minimum of the element numbers from the given structures,
- the *Cosine* measure – it is defined as the number of common elements, divided by the square of sum between the element number from the first structure and the element number from the second structure.

These statistics are generally used for comparing the similarity and diversity of two sample sets, but they can be adapted to our definition couples and their representation. In order to compute a similarity measure between two definitions, each of them are tokenized (segmentation process), lemmatised and syntactic labelled. In this way, a bag of labelled lemmas is obtained for

[2]term frequency-inverse document frequency

each definition. Then, based on the elements of the corresponding bags, the similarity coefficient of two definitions is computed. The considered definitions can be taken from the same dictionary or from different dictionaries (a general one and a specialised one). Based on the obtained similarities we will decide if the two definitions are aligned or not.

By working only with a representation based on these measures, instead of a classical one, the models we propose are able to map the initial vectors (based on a bag of word approach) into a space of reduced dimension, where the computation effort is smaller. Furthermore, we will see if by this reduction we could loose information.

4. SVM alignment

The alignment is considered as a classification problem where each input is represented by the similarity between two definitions. The label associated to that couple of definitions (aligned or not aligned) represents the output. An SVM algorithm [14] is actually used to perform this classification-alignment.

First of all we represent each definition couple by one of the already presented similarity measures. Although it is very simple to work with such representation, we do not know *a priori* which measure works the best. Therefore, we propose to take into account the complementarities between these similarity measures. All five similarity measures are simultaneously considered, obtaining a compact representation for each couple of two definitions. In addition to the similarity measures, the new representation contains the length of each definition too.

The classification process takes place in two phases that reflect the principles of a learning algorithm. Therefore, each data set[3] has to be divided in two parts: a part for training and a part for testing. The training part is divided again in: a learning sub-set – used by the SVM algorithm in order to learn the model that performs the class separation – and a validation sub-set – used in order to optimise the values of the hyper parameters. The SVM model, which is learnt in this manner, classifies (labels) the unseen definition couples from the test set, which is disjoint to the training one.

In order to classify the definition couples, the SVM algorithm uses one of the above representations and a kernel function. The parameters of the SVM model (the penalty for miss-classification C and the kernel parameters) are optimized on the validation set. A cross-validation framework is utilised in order to avoid the over fitting problems. Thus, we automatically adapt the SVM classifier to the problem, actually the alignment of definitions.

[3]that corresponds to all the couples formed by the definitions from two dictionaries

5. Numerical experiments

5.1. **Construction and analysis of the database.** The training database (in fact, definitions aligned by using Coma++) was provided by Yuhan GUO and Rémy DUPAS, IMS - LAPS - GRAI, from University Bordeaux. They have used a set of matchers composed from Affix, 2-gram 3-gram, Edit Distance, Synonym, Soundex and DataType. The weight setting (the weight corresponding to each matcher) was that default. The threshold for accepting an alignment was set to 0.4 (in fact, the couples with the similarity less than the threshold were ignored). Furthermore, the Coma++ tool allows two types of alignment: single condition and multiple conditions. In the first case, only the definitions of concepts have been used in order to perform the alignments, while in the case of a multiple-conditions alignments, Y. Gao and R. Dupas have taken into consideration the definitions, the paths and the fathers of each concept of the two dictionaries.

After a short analysis of the alignments performed by Coma++ we have obtained the following synthesis:

- the number of alignments produced by Coma++ was:
  - mono-condition: 50 pairs of definitions;
  - multi-condition: 159 pairs of definitions;
- the cardinality of alignments in both cases (mono and multi-condition):
  - one to one: a definition WCO was aligned with a single definition CCL;
  - one to many: a CCL definition was aligned with several definitions WCO;
- the alignments cover single and multi-condition:
  - mono-condition $\cap$ multi-condition = 33 couples (in fact, there are 33 common alignments in mono and multi-condition case);
  - mono-condition – multi-condition = 17 couples (17 alignment couples appear in the mono-condition base and they not appear in the multi-condition base);
  - multi-condition – mono-condition = 126 couples (126 alignment couples appear in the multi-condition base and they not appear in the mono-condition base).

5.2. **Numerical experiments performed by SVM.** A set of experiments are performed by using the SVM-based model and the representation discussed in Section 3.3 (that based on five similarity measures).

The train and test data are composed from aligned and not-aligned couples of definitions from CCL and WCO dictionaries, respectively. The aligned couples are represented by the aligned pairs provided by Coma++, while the

not-aligned couples are represented by the definitions from the two dictionaries that were not provided as aligned by Coma++. From all the couples that are formed in this manner, 2/3 of them are considered for training the SVM algorithm and 1/3 of them for testing the aligner.

The C-SVM algorithm, provided by LIBSVM [3], with an RBF kernel is actually used in this experiment. The optimisation of the hyper-parameters is performed by a parallel grid search method. For each combination of these parameters, a 10-fold cross validation[4] is performed during the training phase, the quality of a combination being computed as the average of the accuracy rates estimated for each of the 10 divisions of the data set. Therefore, the best combination is indicated by the best average accuracy rate.

The values of the optimal hyper-parameters and the accuracy rates obtained for 4 different definition processing are presented in Table 1.

Table 1. The performance of the SVM-based aligner.

Pre-processing	RBF Kernel		Accuracy rate	
	γ	C	Mono	Multi
All the words	1/7	10	98.61	98.43
All the words + stemming	1/7	10	98.96	98.07
All the words without stop words	1/7	10	98.26	98.18
All the words without stop words + stemming	1/7	10	98.26	98.07

In order to validate our results, we plan to repeat the experiments by using the model proposed by Ceausu [2], even if a fair comparison between the two models is not possible since the text to be align was different preprocess. We also plan to compare the SVM results with those obtained by other classification algorithm.

6. Conclusions and remarks

In this paper we presented our model for the automatic alignment of definitions taken from two dictionaries (CCL and WCO). The best performances

[4]Cross-validation is a popular technique for estimating the generalization error and there are several interpretations [15]. In k-fold cross-validation, the training data is randomly split into k mutually exclusive subsets (or folds) of approximately equal size. The SVM decision rule is obtained by using $k-1$ subsets on training data and then tested on the subset left out. This procedure is repeated k times and in this manner each subset is used for testing once. Averaging the test error over the k trials gives a better estimate of the expected generalization error.

are obtained by using the SVM algorithm with an RBF kernel and by considering the stemms of all the words of each definition, since the classifier (in fact the hyper-parameters) is better adapted to the alignment task to be solved. However, these conclusions should be validated on some larger corpora.

Further work will be focused on: considering a representation of definitions enriched by semantic and lexical extensions (synonyms, hyponyms, and antonyms) and on developing of an alignment model based on an SVM algorithm with a specialised multiple kernel (this specialisation could be considered in terms of combination of more kernels for text processing (*e.g.* string kernels)).

References

[1] Brown, P. F., Pietra, S. D., Pietra, V. J. D., and Mercer, R. L. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics 19*, 2 (1994), 263–311.

[2] Ceausu, A., Stefanescu, D., and Tufis, D. Acquis communautaire sentence alignment using Support Vector Machines. In *Proceedings of the 5th LREC Conference* (2006), pp. 2134–2137.

[3] Chang, C.-C., and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] Chen, S. F. Aligning sentences in bilingual corpora using lexical information. In *Meeting of the Association for Computational Linguistics* (1993), ACL, pp. 9–16.

[5] Clemons, T. E., and Bradley, E. L. A nonparametric measure of the overlapping coefficient. *Comput. Stat. Data Anal. 34* (2000), 51–61.

[6] Cormen, T., Leiserson, C., and Rivest, R. *Introduction to Algorithms*. MIT Press, 1990.

[7] Dice, L. Measures of the amount of ecologic association between species. *Ecology 26*, 3 (1945), 297–302.

[8] Do, H.-H., and Rahm, E. Coma++ (combination of schema matching approaches), 2010.

[9] Jaccard, P. The distribution of the flora of the alpine zone. *New Phytologist 11* (1912), 37–50.

[10] Moore, R. Fast and accurate sentence alignment of bilingual corpora. In *AMTA '02* (2002), S. D. Richardson, Ed., Springer, pp. 135–144.

[11] Moreau, F., Claveau, V., and Sébillot, P. Automatic morphological query expansion using analogy-based machine learning. In *ECIR 2007* (2007), G.Amati, C. Carpineto, and G. Romano, Eds., vol. 4425 of *LNCS*, Springer, pp. 222–233.

[12] Taskar, B., Lacoste, S., and Klein, D. A discriminative matching approach to word alignment. In *HLT '05* (2005), Association for Computational Linguistics, pp. 73–80.

[13] van Rijsbergen, C., Robertson, S., and Porter, M. New models in probabilistic information retrieval. Tech. Rep. 5587, British Library, 1980.

[14] Vapnik, V. *The Nature of Statistical Learning Theory*. Springer, 1995.

[15] Wahba, G., Lin, Y., and Zhang, H. GACV for Support Vector Machines. In *Advances in Large Margin Classifiers*, B. Smola and S. SchRolkopf, Eds. MIT Press, Cambridge, MA, 1999.

[16] WEI HSU, C., CHUNG CHANG, C., AND JEN LIN, C. A practical guide to support vector classification, 2003.

[1] LITIS, EA - 4108, INSA, ROUEN, FRANCE, [2] COMPUTER SCIENCE DEPARTMENT, BABEŞ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMANIA
E-mail address: adela_sarbu25@yahoo.com, lauras@cs.ubbcluj.ro
E-mail address: arogozan@insa-rouen.fr, pecuchet@insa-rouen.fr