Syntax-Based Concept Alignment for Domain-Specific Machine Translation

Anonymous ACL submission

Abstract

1 Introduction

Grammar-based translation pipelines such as those based on Grammatical Framework (GF) (**?**) have been successfully employed in domain-specific Machine Translation (MT). What makes these systems well suited to the task is the fact that, when we constrain ourselves to a specific domain, where precision is most often more important than coverage, they can provide strong guarantees in terms of grammatical correctness.

Nevertheless, lexical exactness is, in this context, just as important as grammaticality. An important part of the design of the Controlled Natural Language (CNL) the grammar in such a system describes becomes, then, the creation of a translation lexicon. In many cases, this is done for the most part manually, resulting in a time consuming task requiring significant linguistic knowledge. When the grammar is designed based on a parallel corpus of example sentences, it is possible to automate part of this process by means of statistical word and phrase alignment techniques (**?**). None of them is, however, suitable for the common case in which only a limited number of example sentences is available.

In this paper, we propose an alternative approach to the automation of this task. While still being data-driven, our method is grammar-based and, as such, capable of extracting meaningful correspondences even from individual sentence pairs.

A further advantage of performing syntactic analysis is that we do not have to choose *a priori* whether to focus on the word or phrase level. Instead, we can simultaneously operate at different levels of abstraction, thus extracting both single- and multiword, even non-contiguous, correspondences.

For this reason, we refer to the task our system attempts to automate as *Concept Alignment* (CA).

This paper is structured as follows. Section 2 starts by giving an overview of our approach to CA, comparing it to related work, to then focus on our algorithm for extracting correspondences. It is concluded with a description of our method for converting the alignments obtained in this way to a GF translation lexicon. After that, Section 3 presents the results of our first evaluation of the system. Finally, Section 4 consists of a discussion of such results and some ideas for future work.

2 Methodology

The objective of CA is to find semantical correspondences between parts of multilingual parallel texts. We call *concepts* the abstract units of translation, composed of any number of words, identified through this process, and represent them as *alignments*, i.e. tuples of equivalent concrete expressions in different languages.

The basic use case for CA, which we refer to specifically as *Concept Extraction* (CE), is the generation of a translation lexicon from a parallel text. This can be directly compared to the numerous existing word and phrase alignment techniques.

An interesting and less studied variant of CA is *Concept Propagation* (CP), useful for cases where a set of concepts is already known and the goal is to identify the expressions corresponding to each of them in a new language, potentially even working with a different text in the same domain. While our system implements basic CP functionalities, however, in this paper we focus on CE and restrict its application to bilingual corpora.

As stated in the Introduction, most existing extraction solutions are based on statistical approaches and are, as a consequence, unsuitable for small datasets. Grammar-based approaches, making use of parallel treebanks and collectively referred to as *tree-to-tree alignment methods*, have also been proposed (?), but have historically suffered from the inconsistencies between the various constituency grammar formalisms used to define grammars for the different languages and from the insufficient degree of robustness of existing parses.

This work is a new attempt in the same direction, enabled by two multilinguality-oriented grammar formalisms developed over the course of the last 25 years: Universal Dependencies (UD) (?) and Grammatical Framework (GF) (?).

While both formalisms, the former a dependency and the latter a constituency grammar, independently solve the former issue, UD is especially appealing since dependency trees are an easier target for parsing. As a consequence, several robust dependency parsers, such as (?) and (?) are available.

UD parsing alone would then be sufficient to extract (or propagate) tree-to-tree alignments, but not to automate the generation of a ready-to-use, morphologically-aware translation lexicon. This is where GF comes into play: after correspondences are inferred from a parallel text, our proposed system is able to convert them to GF grammar rules, easy to embed in a domain-specific grammar but also making it possible to immediately evaluate the system by carrying out small-scale translation experiment using pre-existing grammatical constructions implemented in GF's Resource Grammar Library (RGL) (?). This is made possible by `gf-ud`, a conversion tool described in (?) and (?).

Concretely, the system we propose requires the following elements, whose reciprocal relations are shown in Figure 1:

- a UD parser
- an alignment module based on dependency tree comparison
- a program, based on `gf-ud`, that converts the alignments into GF grammar rules and uses them to construct a translation lexicon

2.1 Extracting concepts

The core part of the system outlined above is of course the alignment module. In this section, we present our method for extracting alignments from parallel bilingual UD treebanks, which can be obtained from sentence-aligned texts with any UD parser.

The basic extraction algorithm, whose pseudocode is shown in Figure **??**, takes as input a list of priority-sorted *alignment criteria*, rules to determine whether two dependency trees should be aligned with each other which we will discuss in greater detail in Section 2.1.1, and a pair of UD trees corresponding to the sentences to align. From an implementation point of view, UD trees are, as in `gf-ud`, rose trees where each node represents a token, obtained from the CoNLL-U files produced by the UD parser.

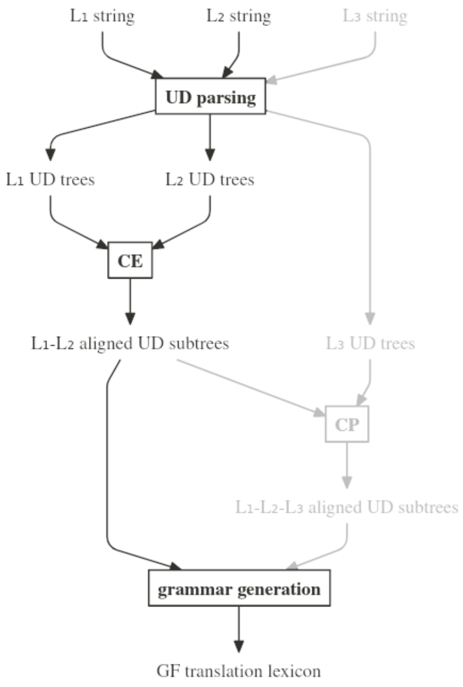

Figure 1: System overview. Parts in grey are currently at the prototype stage and will not be further discussed in this paper.

As a first step, the program checks whether the two full sentence trees can be aligned with each other, i.e. if they match any of the alignment criteria. If this is the case, they are added to a collection of alignments, also represented as a pair of UD trees associated with some metadata, which is what the function will return after aligning all the dependency subtrees.

Depending on which alignment criteria they match, the head of the two trees may or may not also be added to such collection. This is important...

The same procedure is applied recursively to all pairs of immediate subtrees of each sentence, until the leaves are reached or alignment is no longer possible due to lack of matching criteria.

When working on corpora consisting of multiple sentences, this algorithm can be applied in an iterative fashion, so that knowledge gathered when a sentence pair is aligned can be reused when working on the following ones and to keep track of the number of occurrences of each alignment throughout the entire text.

2.1.1 Alignment criteria

While alignment criteria are customizable, to better understand the extraction algorithm described in the above we dedicate this section to describing the ones that are currently used by default.

Matching UD labels The most obvious, but also most effective idea is to determine alignability based on comparing the dependency labels of the candidate UD tree pair. In particular, according to this idea, two subtrees, in *matching context*, i.e. attached to aligned heads constitute an alignment if their roots share the same dependency label. Intuitively, this means that they are in the same relation with their heads.

Note that, since the root of US trees is always attached to a a fake node with an arc labelled `root`, this criterion also makes it so that full sentences are always considered to align with each other, something which is desirable since we assume the parallel texts that are fed to our program to be sentence-aligned.

Part-Of-Speech equivalence CoNNL-U files provide information not only on the syntactic role of the tokens a sentence is composed of, but also on their grammatical categories, represented as Universal Part-Of-Speech (POS) tags. Intuitively, if the words corresponding to the nodes two trees in matching contexts are composed of have the same POS tags, those two trees are generally more likely to correspond to each other than if not. As a consequence, a useful relation to define between dependency trees is that of *POS-equivalence*:

Definition *Two dependency trees t, u are* POS-equivalent (and, as such, will be aligned) if $M_1 = M_2 \neq \emptyset$, where M_i is defined as the multiset of POS tags of all the meaning-carrying *(see below) word nodes of t_i.*

The definition specifies that the two multisets should not contain the POS tags of all the words in the sentence but rather only those of the *meaning-carrying* ones, referring to words belonging to a particular set of classes. Words that should generally be taken into account, in fact, correspond roughly to content words (cf. Section **??**), but this term was deliberately avoided in Definition **??** due to the fact that it can be useful also to include some function words, for instance pronouns and some kinds of determiners. In particular, the current implementation considers as meaning-carrying all words that belong to an open class - defined as in the UD documentation (**?**) - and numerals, but this set of tags has been obtained empirically working with English-Italian and might not be ideal for all language pairs. On the other hand, words belonging to other classes, such as auxiliary verbs, adpositions and conjunctions can and should be in most cases ignored as they are often omitted or rendered with words with different POS tags when the sentence is translated to another language, especially if the two languages in the pair at hand differ significantly.

Applied alone, this criterion can be used to capture correspondences that would otherwise be missed, thus increasing recall, but a decrease in precision is also to be expected. Perhaps more interestingly, however, another way to apply this criterion is in conjunction with other ones, and in particular together with UD label matching (cf. Criterion **??**), in context where high precision is more important than recall. We will get back to combining criteria in Section **??**.

Known translation divergence

Known alignment

2.1.2 Pattern matching

2.2 Generating grammar rules

3 Evaluation

4 Conclusions