Adding syntactic structure to bilingual terminology for improved domain adaptation

Mikel Artetxe[1], Gorka Labaka[1], Chakaveh Saedi[2], João Rodrigues[2], João Silva[2], António Branco[2], Eneko Agirre[1]
[1] IXA Group, Faculty of Computer Science, University of the Basque Country, Spain
[2] Department of Informatics, Faculty of Sciences, University of Lisbon, Portugal
[1] {mikel.artexe, gorka.labaka, e.agirre}@ehu.eus,
[2] {chakaveh.saedi, joao.rodrigues, jsilva, antonio.branco}@di.fc.ul.pt

Abstract

Deep-syntax approaches to machine translation have emerged as an alternative to phrase-based statistical systems. TectoMT is an open source framework for transfer-based MT which works at the deep tectogrammatical level and combines linguistic knowledge and statistical techniques. When adapting to a domain, terminological resources improve results with simple techniques, e.g. force-translating domain-specific expressions. In such approaches, multiword entries are translated as if they were a single token-with-spaces, failing to represent the internal structure which makes TectoMT a powerful translation engine. In this work we enrich source and target multiword terms with syntactic structure, and seamlessly integrate them in the tree-based transfer phase of TectoMT. Our experiments on the IT domain using the Microsoft terminological resource show improvement in Spanish, Basque and Portuguese.

1 Introduction

TectoMT (Žabokrtský et al., 2008; Popel and Žabokrtský, 2010) has emerged as an architecture to develop deep-transfer systems, where the translation step is done a deep level of analysis, in contrast to methods based on surface sequences of words. TectoMT combines linguistic knowledge and statistical techniques, particularly during transfer, and it aims at transfer on the so-called tectogrammatical layer (Hajičová, 2000), a layer of deep syntactic dependency trees.

In domain adaptation of machine translation, a typical scenario is as follows: there is an MT system trained on large general-domain data, and there is a bilingual terminological resource which covers part of the vocabulary of the target domain. In this case, a simple force-translate approach can suffice to obtain good results (Dušek et al., 2015). In the context of TectoMT, this approach is implemented identifying source terms in the analysis phase, and adding as a single node in the tree. In the case of multiword terms, this means that the internal structure is not captured and that it is not possible to access the internal morphological and syntactic information.

In this work we enrich source and target multiword terms with syntactic structure (so-called "treelets"), and seamlessly integrate them in the tree-based transfer phase of TectoMT. This allows to check for morphological agreement when producing translation (e.g. gender of noun-adjective terms in Spanish). The results on three languages within the Information Technology (IT) domain show consistent improvements when applied on the Microsoft terminological resource.

2 TectoMT

As most rule-based systems, TectoMT consists of analysis, transfer and synthesis stages. It works on different levels of abstraction up to the tectogrammatical level (cf. Figure 1) and uses *blocks* and *scenarios* to process the information across the architecture (see below).

2.1 Tecto layers

TectoMT works on an stratified approach to language, that is, it defines four layers in increasing level of abstraction: raw text (w-layer), morphological layer (m-layer), shallow-syntax layer (a-layer), and

Proceedings of the 2nd Deep Machine Translation Workshop (DMTW 2016), pages 39–46,
Lisbon, Portugal, 21 October 2016.

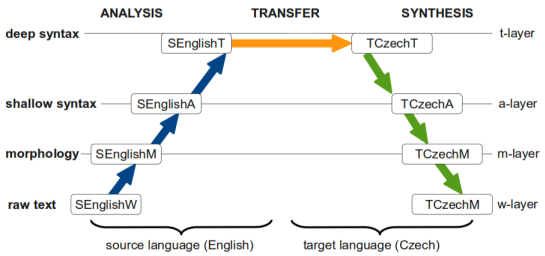

Figure 1: The general TectoMT architecture (from Popel and Žabokrtský (2010:298)).

deep-syntax layer (t-layer). This strategy is adopted from the Functional Generative Description theory (Sgall, 1967), further elaborated and implemented in the Prague Dependency Treebank (PDT) (Hajič et al., 2006). As explained by Popel and Žabokrtský (2010:296), each layer contains the following representations (see Figure 2):

Morphological layer (m-layer) Each sentence is tokenized and tokens are annotated with a lemma and morphological tag, e.g. *did: do-VBD*.

Analytical layer (a-layer) Each sentence is represented as a shallow-syntax dependency tree (a-tree), with a 1-to-1 correspondence between m-layer tokens and a-layer nodes. Each a-node is annotated with the type of dependency relation to its governing node, e.g. *did* is a dependent of *tell (VB)* with a *AuxV* relation type.

Tectogrammatical layer (t-layer) Each sentence is represented as a deep-syntax dependency tree (t-tree) where lexical words are represented as t-layer nodes, and the meaning conveyed by function words (auxiliary verbs, prepositions and subordinating conjunctions, etc.) is represented in t-node attributes, e.g. *did* is no longer a separate node but part of the lexical verb-node *tell*. The most important attributes of t-nodes are:

tectogrammatical lemma;

functor the semantic value of syntactic dependency relations, e.g. actor, effect, causal adjuncts;

grammatemes semantically oriented counterparts of morphological categories at the highest level of abstraction, e.g. tense, number, verb modality, negation;

formeme the morphosyntactic form of a t-node in the surface sentence. The set of formeme values depends on its semantic part of speech, e.g. noun as subject (n:subj), noun as direct object (n:obj), noun within a prepositional phrase (n:*in*+X) (Dušek et al., 2012).

2.2 The TectoMT system

TectoMT is integrated in Treex,[1] a modular open-source NLP framework. Blocks are independent components of sequential steps into which NLP tasks can be decomposed. Each block has a well-defined input/output specification and, usually, a linguistically interpretable functionality. Blocks are reusable and can be listed as part of different task sequences. We call these *scenarios*.

TectoMT includes over 1,000 blocks; approximately 224 English-specific blocks, 237 for Czech, over 57 for English-to-Czech transfer, 129 for other languages and 467 language-independent blocks.[2] Blocks vary in length, as they can consist of a few lines of code or tackle complex linguistic phenomena.

3 Terminology as Gazetteers

The easiest form to exploit domain terminology is to use them as fixed translation units, where the term needs to appear in the source text in a fixed inflectional form. That is, if the form appears in

[1] `https://ufal.mff.cuni.cz/treex`
[2] Statistics taken from: `https://github.com/ufal/treex.git (27/08/2015)`

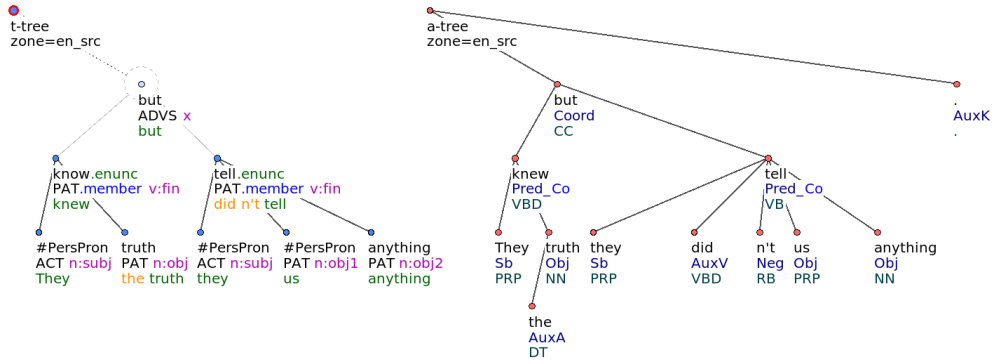

Figure 2: a-level and t-level English analysis of the sentence "`They knew the truth but they didn't tell us anything.`"

English		Spanish	
`liboff_1`	`Accessories`	`liboff_1`	`Accesorios`
`liboff_2`	`Start at`	`liboff_2`	`Empezar en`
`kde_1`	`Programs`	`kde_1`	`Programas`
`kde_2`	`System tools`	`kde_2`	`Herramientas del sistema`
`kde_3`	`Start`	`kde_3`	`Iniciar`
`kde_4`	`Disk`	`kde_4`	`Disco`
`kde_5`	`PC running on low battery`	`kde_5`	`Equipo funcionando bajo de bateria`
`kde_6`	`System`	`kde_6`	`Systema`
`kde_7`	`Start`	`kde_7`	`Comenzar`
`wiki_1`	`PC`	`wiki_1`	`PC`

Figure 3: A sample of English-Spanish terminological resources from localization files.

some inflected form which is not present in the dictionary, it is not translated. Given that terminological resources contain mainly base forms, several terms are missed in the source texts. The property of having a fixed form allows for easy implementation: match the source expression in the terminological resource in the source text and replace it deterministically by its equivalent.

In this work we are interested in the IT domain, concerning software texts which includes, among other, menu items, button names, sequences of those and system messages.

3.1 Lexicon collection and format

The straightforward way to obtain terminology resources is to extract them from freely available software localization files. We designed a general extractor that accepts `.po` localization files and outputs a lexicon. The lexicon is formed by two lists containing corresponding expressions in two languages. Each of the two lists consist of two columns: a unique expression identifier, the expression itself. The identifier is the same for equivalent terms. Figure 3 shows an excerpt from an English-Spanish gazetteer.

3.2 Translation method

Translation using gazetteers proceeds in multiple steps:

Matching the lexicon items. This is the most complex stage of the whole process. It is performed just after the tokenization, before any linguistic processing is conducted. Lexicon items are matched in the source tokenized text and the matched items, which can possibly span several neighboring tokens, are replaced by a single-word placeholder.

In the initialization stage, the source language part of the lexicon is loaded and structured in a word-based trie to reduce time complexity of the text search. In the current implementation, if an expression appears more than once in the source gazetteer list, only its first occurrence is stored. Therefore, the performance of gazetteer matching machinery depends on the ordering of the gazetteer lists. A trie built

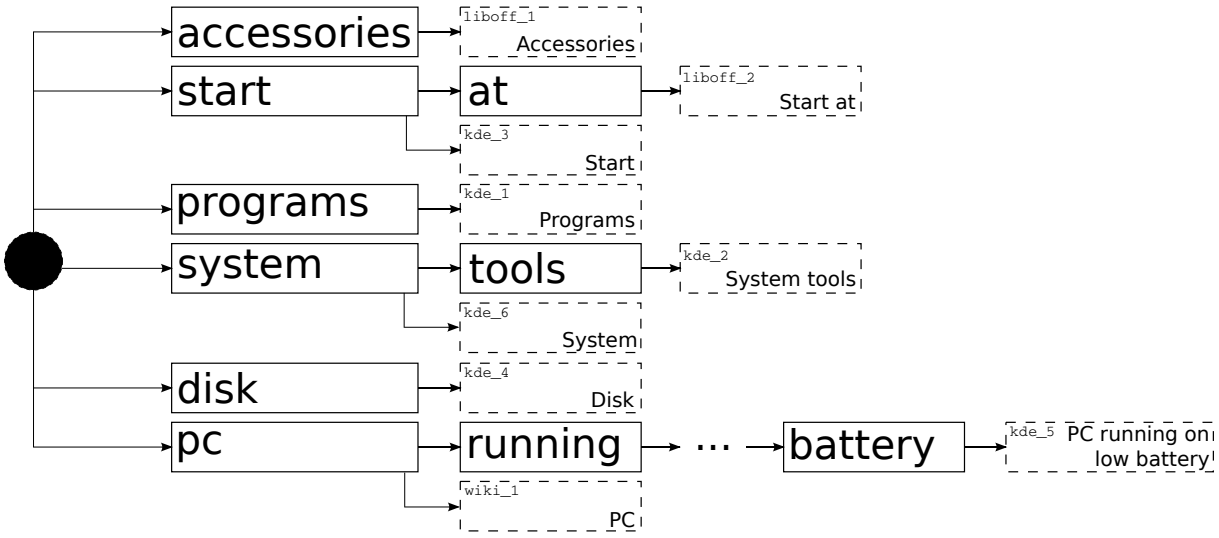

Figure 4: A trie created from the English terms in Figure 3

from the English list of the sample English-Spanish gazetteer is depicted in Figure 4. Note that the `kde_7` item is not represented in the trie, since the slot is already occupied by the `kde_3` item.

The trie is then used to match the expressions in the list to the source text. The matched expressions might overlap. A scoring function estimates whether the term is actually a term in the text. Thus, every matched expression is assigned a score. entity. Figure 5 shows a sample sentence (a), including matched expressions and scores assigned (b). The matches with positive score are ordered by the score and filtered to get non-overlapping matches, taking those with higher score first. The matched words belonging to a single term are then replaced by a single placeholder word (see Figure 5c).

As a last step, the neighboring terms are collapsed into one and replaced by the placeholder word. As a heuristic for the IT domain, terms that occur separated by a $>$ symbol are also collapsed. This measure is aimed at translation of menu items and button labels sequences, which frequently appear in the IT domain corpus. After this step, the sample sentence becomes drastically simplified, which should be much easier to process by a part-of-speech tagger and parser (see Figure 5e). However, all the information necessary to reconstruct the original expressions or their lexicon translations are stored (see Figure 5d).

Translating matched items. The expressions matched in the source language are transferred over the tectogrammatical layer to the target language. Here, the placeholder words are substituted by the expressions from the target language list of the gazetteer, which are looked up using the identifiers coupled with the placeholder words. Possible delimiters are retained. This is performed before any other words are translated. The tectogrammatical representation of the simplified sample English sentence (Figure 5d) is transferred to Spanish by translating the gazetteer matches first, followed by the standard TetoMT steps (lexical choice for the other words and concluded with the synthesis stage, cf. Figure 5g).

4 Terminology as treelets

As shown in the previous section, simple string matching with gazetteers is appropriate to translate fixed terms in the IT domain like menu items, button names and system messages. However, this technique has two important limitations when applied to terminology other than those fixes terms, including common nouns (driver, file...) or verbs (run, set up...):

1. It does not handle inflection, neither in the source language nor in the target language, so the different surface forms of a given term (e.g. run, runs, running, ran) will not be translated unless there is a separate entry for each of them. This is particularly relevant for morphologically rich languages like Spanish (verb inflection) or Basque.

a) To defragment the PC, click Start > Programs > Accessories > System Tools > Disk Defragment.

b) To defragment the [PC wiki_1=24], click [Start kde_3=24] > [Programs kde_1=24] > [Accessories liboff_1=24] > [[System kde_6=24] Tools kde_2=44] > [Disk kde_4=24] Defragment.

c) To defragment the [PH wiki_1], click [PH kde_3] > [PH kde_1] > [PH liboff_1] > [PH kde_2] > [PH kde_4] Defragment.

d) To defragment the [PH wiki_1], click [PH kde_3 > kde_1 > liboff_1 > kde_2 > kde_4] Defragment.

e) To defragment the PH, click PH Defragment.

f) To defragment the [PC wiki_1], click [Comienzo > Programas > Accesorios > Herramientas del sistema > Disco kde_3 > kde_1 > liboff_1 > kde_2 > kde_4] Defragment.

g) Desfragmentador el PC haga clic Iniciar > Programas > Accesorios > Herramientas del Sistema > Disco desfragmentador.

Figure 5: A sample English sentence processed by the English-Spanish gazetteer. Translation process is shown step by step. See text for details. PH stands for placeholder

2. It does not handle morphosyntactic ambiguity. For instance, the English term “test” can either be a noun or a verb, and its translation depends on that.

In order to overcome these issues, we developed a terminology translation module which is applied on the t-layer. The translation process involves the following steps:

1. **Preprocessing:** The terminology dictionary is first preprocessed so it can be efficiently used later at runtime. For that purpose, the lemma of each entry in the dictionary is independently analyzed up to the t-layer in both languages. This analysis is done without any context, so if there is some ambiguity, it might happen that the analysis given by the system does not match the sense it has in the dictionary. For instance, the English term ’file’ might be analyzed either as a verb or a noun, but its entry in the dictionary and, consequently, its translation, will correspond to only one of these senses. For that reason, we decide to remove all entries whose part-of-speech tag in the original dictionary does not match the one assigned to the root node by the analyzer.

2. **Matching:** During this stage, we search for occurrences of the dictionary entries in the text to translate, which is done at the t-layer. For that purpose, the preprocessed tree of a term is considered to match a subtree of the text to translate if the lemma and part-of-speech tag of their root node are the same and their corresponding children nodes recursively match for all their attributes. By limiting the matching criteria of the root node to the lemma and part-of-speech, the system is able to match different surface forms of a single entry (e.g. “local area network” and “local area networks”). Note that, thanks to the deep representation used at the t-layer, we are also able to capture form variations in tokens other than the root. For instance, in Spanish both adjectives and nouns carry gender and number information, but in the t-layer only the highest node encodes this information. This way, the system will be able to match both “disco duro” (“hard disk”) and “discos duros” (“hard disks”) for a single dictionary entry, even if the surface form of the children node “duro” was not the same in the original text. In addition to that, it should be noted that we do allow the subtree of the text to translate to have additional children nodes to the left or right, but only at the

	en-eu	en-es	en-pt
KDE	70,298	98,510	98,505
LibreOffice	70,991	75,482	75,743
VLC	5,548	6,214	6,215
Wikipedia	1,505	24,610	20,239
Total Localization	148,342	204,816	200,702
Microsoft Terminology	6,474	25,069	15,748

Table 1: Source and number of gazetteer entries in each language.

first level below the root node, so we are able to match chunks like "corporate local area network" or "external hard disk" for the previous examples.

In order to do the matching efficiently, we use a prebuilt hash table that maps the lemma and part-of-speech pair of the root node of each dictionary entry to the full tree obtained in the preprocessing stage. This way, for each node in the input tree, we look up its lemma and part-of-speech in this hash map and, for all the occurrences, recursively check if their children nodes match.

3. **Translation:** During translation, we replace each matched subtree with the tree of its corresponding translation in the dictionary, which was built in the preprocessing stage. For that purpose, the children nodes of the matched subtree are simply removed and the ones from the dictionary are inserted in their place. As for the root node, the lemma and part-of-speech are replaced with the one from the dictionary, but all the other attributes are left unchanged. Given that these attributes are language independent, the appropriate surface form will then be generated in subsequent stages, so for our example "local area network" is translated as "red de area local" while "local area networks" is translated as "redes de area local", even if there is a single entry for them in the dictionary.

5 Experiments

We conducted experiments in three languages, using English as the source language. The experiments were carried on an IT dataset released by the QTLeap project.[3] The systems were trained in publicly available corpora, mostly Europarl, with the exception of Basque, where we used an in-house corpus for training.

Localization Gazetteers The gazetteers for Basque, Spanish and Portuguese were collected from four different sources: the localization files of VLC,[4] LibreOffice,[5] and KDE[6]; and IT-related Wikipedia articles. In addition, some manual filtering (blacklisting) was performed on all the gazetteers.

For mining IT-related terms from Wikipedia, we adopted the method by Gaudio and Branco (2012). This method exploits the hierarchical structure of Wikipedia articles. This structure allows for extracting articles on specific topics, selecting the articles directly linked to a superordinate category. For this purpose, Wikipedia dumps from June 2015 were used for each of the languages, and they were accessed using the Java Wikipedia Library, an open-source, Java-based application programming interface that allows to access all information contained in Wikipedia (Zesch et al., 2008). Using as starting point the most generic categories in the IT field, all the articles linked to these categories and their children were selected. The titles of these article were used as entries in the gazetteers. The inter-language links were used to translate the title in the original languages to English. Similar result could be expected if the method was applied to the Linked Open Data version of Wikipedia, DBPedia,

[3] More specifically on the Batch2 answer corpus

[4] `http://downloads.videolan.org/pub/videolan/vlc/2.1.5/vlc-2.1.5.tar.xz`

[5] `http://download.documentfoundation.org/libreoffice/src/4.4.0/libreoffice-translations-4.4.0.3.tar.xz`

[6] `svn://anonsvn.kde.org/home/kde/branches/stable/l10n-kde4/{es,eu,pt}/messages`

	en→es
TectoMT	29.60
+Gazetteers	32.01
+Gazetteers+Msoft$_{Gazetteer}$	32.25
+Gazetteers+Msoft$_{Treelet}$	34.16

Table 2: BLEU scores for Spanish

	en→eu	en→pt
TectoMT	17.15	21.96
+Gazetteers	20.51	22.68
+Gazetteers+Msoft$_{Treelet}$	23.41	23.01

Table 3: BLEU scores for Basque and Portuguese

The figures of collected gazetteer entries for all the sources are presented in Table 1. The gazetteers have been released through Meta-Share.[7]

Microsoft Terminology Collection The Microsoft Terminology Collection is publicly available for nearly 100 languages[8]. It uses the standard TermBase eXchange (TBX) format and, for each entry, it includes the English lemma, the target language lemma, their part-of-speech in both language, and a brief definition in English. Note that the dictionary also includes many multiword terms, such as "local area network" or "single click".

5.1 Results for Spanish

The results in Table 2 show the results of the two baselines: TectoMT without gazetteers and TectoMT with all gazetteers, except the Microsoft gazetteer. When including the Microsoft terminology as a gazetteer, there is a small improvement. When including the Microsoft terminology as treelets, the improvement is larger, up to 34.16.

5.2 Results for Basque and Portuguese

Given the good results, we repeated a similar experiment for Basque and Portuguese (cf. Table 2). We also show the results of the two baselines: TectoMT without gazetteers and TectoMT with all gazetteers, except the Microsoft gazetteer. When including the Microsoft terminology as treelets, we also obtain an improvement in both languages, larger for Basque and smaller for Portuguese.

6 Conclusions

In this paper we present a system for terminology translation based on deep approaches. We analyse the terms in the resource, and integrate them in a deep syntax-based MT engine, TectoMT. Our method is able to translate complex terms exhibiting different morphosyntactic agreement phenomena. The results on the IT domain show that this method is effective for Spanish, Basque and Portuguese when applied on the Microsoft terminological resource. For the future, we would like to extend our approach to the rest of the terminological resources, and to present more experiments and error analysis to show the value of our approach.

Acknowledgements

The research leading to these results has received funding from FP7-ICT-2013-10-610516 (QTLeap) and from P2020-3279 (ASSET).

[7] http://metashare.metanet4u.eu/go2/qtleap-specialized-lexicons
[8] https://www.microsoft.com/Language/en-US/Terminology.aspx

References

Ondřej Dušek, Zdeněk Žabokrtský, Martin Popel, Martin Majliš, Michal Novák, and David Mareček. 2012. Formemes in English-Czech deep syntactic MT. In *Proceedings of the Seventh Works-hop on Statistical Machine Translation*, pages 267–274. Association for Computational Linguistics.

Ondřej Dušek, Luís Gomes, Michal Novák, Martin Popel, and Rudolf Rosa. 2015. New language pairs in tectomt. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 98–104, Lisbon, Portugal, September. Association for Computational Linguistics.

Rosa Gaudio and Antonio Branco. 2012. Using wikipedia to collect a corpus for automatic definition extraction: comparing english and portuguese languages. In *Anais do XI Encontro de Linguistiva de Corpus - ELC 2012*, Instituto de Ciłncias Matemticas e de Computao da USP, em So Carlos/SP.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štepánek, Jiří Havelka, Marie Mikulová, Zdenek Žabokrtský, and Magda Ševcıková Razımová. 2006. Prague dependency treebank 2.0. *CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia*, 98.

Eva Hajičová. 2000. Dependency-based underlying-structure tagging of a very large Czech corpus. *TAL. Traitement automatique des langues*, 41(1):57–78.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: modular NLP framework. In *Advances in natural language processing*, pages 293–304. Springer.

Petr Sgall. 1967. Functional sentence perspective in a generative description. *Prague studies in mathematical linguistics*, 2(203-225).

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170. Association for Computational Linguistics.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wikictionary. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco. European Language Resources Association (ELRA).