ZAP: An Open-Source Multilingual Annotation Projection Framework

Alan Akbik and **Roland Vollgraf**
Zalando Research
Mühlenstraße 25, 10243 Berlin
{firstname.lastname}@zalando.de

Abstract

Previous work leveraged *annotation projection* as a convenient method to automatically generate linguistic resources such as treebanks or propbanks for new languages. This approach automatically transfers linguistic annotation from a resource-rich source language (SL) to translations in a target language (TL). However, to the best of our knowledge, no publicly available framework for this approach currently exists, limiting researchers' ability to reproduce and compare experiments. In this paper, we present ZAP, the first open-source framework for annotation projection in parallel corpora. Our framework is Java-based and includes methods for preprocessing corpora, computing word-alignments between sentence pairs, transferring different layers of linguistic annotation, and visualization. The framework was designed for ease-of-use with lightweight APIs. We give an overview of ZAP and illustrate its usage.

The framework is available on github at https://github.com/zalandoresearch/zap

Keywords: Annotation projection, corpora, multilingual data

1. Introduction

Linguistically annotated corpora, such as *treebanks* (Marcus et al., 1993) or *propbanks* (Palmer et al., 2005), are a crucial driver of progress in natural language processing research. As a cost-effective alternative to manual annotation, previous work explored the use of *annotation projection* (Yarowsky et al., 2001) in parallel corpora to automatically create linguistically annotated corpora for new languages. This approach requires only a parallel corpus consisting of sentences in a resource-rich source language (SL) and their translations in a target language (TL), as well as existing parsers for the SL. It leverages the hypothesis that translated sentences share a degree of syntactic and, in particular, semantic parallelism (Padó and Lapata, 2009), thus allowing us to automatically transfer linguistic annotations from SL to TL. Refer to Figure 1 for an illustration of this approach, and Section 2 for more details.

Annotation projection resources. Previous works leveraged annotation projection for various layers of syntactic and semantic annotation (Yarowsky et al., 2001; Hwa et al., 2005; Van der Plas et al., 2011; Akbik et al., 2015). However, to the best of our knowledge, no publicly available framework exists for this approach. This limits the ability of researchers to quickly set up experiments, discuss and compare approaches against previous work, and analyze the viability of annotation projection for a specific type of linguistic annotation or language pair.

ZAP framework. For these reasons, we present ZAP, the first open-source framework for annotation projection. The framework contains a number of components callable through lightweight APIs:

1. A set of preprocessing tools that wrap popular NLP libraries to facilitate parsing of sentences in all supported languages.

2. An alignment module for word-aligning translated sentence pairs in a parallel corpus.

3. An annotation projection module for transferring different types of linguistic annotation between word-aligned sentences.

4. A visualization module for rendering word-aligned sentence pairs and projected annotations, enabling researchers to execute and visually inspect annotation projection for specific sentence pairs.

In this extended abstract, we first give a brief overview of annotation projection and related work. We then introduce the ZAP framework and present a walkthrough of core API calls and functionality. Finally, we discuss extensibility of the framework and future work.

2. Annotation Projection

We first briefly illustrate the principle of annotation projection using the sentence pair in Figure 1. Here, the source language is English and the target language is German. Our goal is to automatically generate linguistic annotation for the German sentence. In the first step (Figure 1.a), we use syntactic and semantic parsers to predict part-of-speech (PoS) tags, dependencies and semantic roles for the English sentence. We also perform word alignment to link each English word to its German translation.

We then successively transfer linguistic annotation along these word alignments to the German sentence. We begin with word-level PoS tags (see Figure 1.b). For instance, we transfer the NOUN tags from the English words *cat* and *cheese* onto the German translations *Katze* and *Käse*, thereby marking them up as nouns. We then also project dependency arcs and labels, as well as semantic roles (see Figure 1.c), thus learning that *Katze* (like the *cat* in the SL) is a syntactic subject that takes the semantic role of CONSUMER in the target sentence. Thus, the German sentence is automatically annotated with multiple layers of linguistic annotation.

Previous work. Previous work used annotation projection for a wide range of annotations, including PoS tags

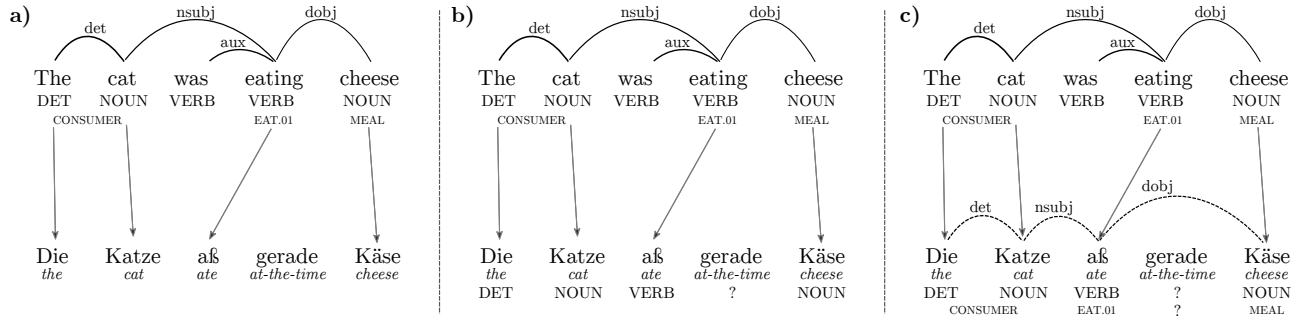

Figure 1: Stepwise example of annotation projection for an English-German sentence pair. In a), SL parsers are used to predict syntax and semantic roles for the English sentence, and words are aligned to their German translations. In b) PoS tags are transferred along alignments onto German words. In c) dependency arcs and semantic roles are transferred along word alignments, thus automatically labeling the German sentence with dependencies, PoS tags and semantic roles. Note that the unaligned German word *gerade* remains unannotated.

(Yarowsky et al., 2001), syntactic chunks (Yarowsky and Ngai, 2001), dependency trees (Hwa et al., 2005), word senses (Bentivogli and Pianta, 2005), named entities (Mayhew et al., 2017) and semantic roles (Padó and Lapata, 2009; Akbik et al., 2015).

However, as the example in Figure 1 shows, annotation projection may not always produce fully annotated target language sentences. Previous work found annotation projection to be sensitive to factors such as the quality of translations, accuracy of word alignments and parsers (Akbik et al., 2015), and subject to errors stemming from translational divergences (Dorr, 1994). For these reasons, previous works devised a number of strategies to address such issues. These include identifying and filtering suboptimally aligned sentence pairs from the parallel corpus (Ni et al., 2017), blocking the projection of specific annotations (Van der Plas et al., 2011), guiding projection using cross-lingual statistics (Täckström et al., 2012), using semi-supervised learning to fill annotation gaps in the target language corpus (Akbik et al., 2015) and leveraging crowdsourcing to curate projections (Wang et al., 2017).

With the release of ZAP, we aim to assist researchers investigate and further improve such annotation projection strategies for various layers of linguistic annotation, with the hope of eventually generating new linguistic resources for low-resource languages that approach the quality of expert annotation.

3. The ZAP Framework

The ZAP framework is a Java-based open source project available on github[1]. It provides both a set of lightweight APIs to programmatically design annotation projection experiments, as well as a simple UI for exploratory analysis of projection approaches. In the following, we illustrate the core usage of the framework.

3.1. Packaging and Distribution

The project is managed using the *maven* build automation tool, giving researchers two simple ways to start using the framework. The first is to clone the github repository and build the project locally by calling `mvn install` in its root folder, which will execute unit tests and install the project into the local maven repository. This is the recommended option for researchers interested in fully understanding and extending the APIs for their experiments. The second option is to include the project as a dependency into a Java project. This is accomplished simply by adding the following dependency to the project's POM.XML:

```
<dependency>
 <groupId>org.zalando.research</groupId>
 <artifactId>zap</artifactId>
 <version>1.0</version>
</dependency>
```

We recommend the latter option for most researchers, allowing them to quickly get started with building their own projection experiments.

3.2. Parser Wrappers

The first component of ZAP are wrappers around popular NLP tools, namely STANFORD CORENLP (Manning et al., 2014) (for PoS-tagging, named entity recognition and dependency parsing), the ANNA lemmatizers (Bohnet, 2010) (for lemmatization) and the MATE toolkit (Björkelund et al., 2009) (for semantic role labeling). These tools were selected for their open source availability and their maven packaging. This design choice ensures that users do not need to install any third party parsers or tools to get started – maven downloads all required dependencies automatically.

Parsing a sentence. After adding ZAP to a project, users can parse any sentence with two lines of code:

```
PipelineWrapper pipeline = new
    PipelineWrapper(Language.ENGLISH);

Sentence parse = pipeline.parse("The
    cat was eating cheese.");
```

The first line initializes the parser wrapper for the selected language, the second parses the sentence into an object that contains the full syntactic-semantic parse of the sentence.

[1] https://github.com/zalandoresearch/zap

This object provides methods for accessing annotations and rendering itself in *CoNLL-U*[2] format.

Different languages. In order to parse a sentence in a different language, the wrapper only needs to be initialized with the appropriate language enum, as follows:

```
PipelineWrapper pipeline = new
    PipelineWrapper(Language.GERMAN);

Sentence parse = pipeline.parse("Die
    Katze aß gerade Käse.");
```

At time of writing, ZAP wraps tools for parsing of English, German, French, Spanish and Chinese. We expect more languages added to this list in the near future.

Gold-annotated corpora. It is also possible to create SENTENCE objects from existing gold-annotated treebanks in lieu of using parsers. For this purpose, we include classes for reading corpora in CoNLL-U and CoNLL-X format.

3.3. Heuristic Word Alignment

A prerequisite for annotation projection is to word-align parallel sentences. A first step is to create a BISENTENCE object that contains a source sentence and its target language translation. The source sentence is typically parsed (as described above), while the target sentence is initialized without annotation:

```
Sentence sourceSentence =
    pipeline.parse("The cat was eating
    cheese.");

Sentence targetSentence =
    Sentence.fromTokenized("Die Katze
    aß gerade Käse.");

BiSentence biSentence = new
    BiSentence(sourceSentence,
    targetSentence);
```

The ZAP framework offers several ways to add word alignments to a BISENTENCE object. One way is to read externally computed word alignments in the "Pharaoh format", as produced by most popular word-alignment tools including FASTALIGN (Dyer et al., 2013) and BERKELEYALIGNER (DeNero and Liang, 2007). As an alternative option, ZAP also offers a heuristic word alignment module that uses pre-computed word translation probabilities computed over large parallel corpora (Tiedemann, 2012). The latter option has the advantage of not requiring external tools. It can be called by instantiating the HEURISTICALIGNER.

```
HeuristicAligner aligner =
    HeuristicAligner
  .getInstance(Language.GERMAN);

biSentence.align(aligner);
```

[2] `http://universaldependencies.org/format.html` lists a specification of the format.

```
1   Die     _   DET   DT   _   2   det
2   Katze   _   NOUN  NN   _   3   nsubj
3   aß      _   VERB  VBG  _   0   root
4   gerade  _   _     _    _   0   _
5   Käse    _   NOUN  NN   _   3   dobj
6   .       _   _     _    _   0   _
```

Figure 2: German example sentence with projected annotations rendered in CoNLL-U format (first 8 columns).

3.4. Annotation Projection

Once a word-aligned BISENTENCE is produced, a simple call of the ANNOTATIONTRANSFER object suffices to project all word-level (PoS-tags), span-level (named entities), tree-level (dependencies) and proposition semantic levels of annotation from the source sentence onto the target sentence:

```
new AnnotationTransfer()
    .transfer(biSentence);
```

In ZAP, annotation is projected following standard practices. Word-level annotation is projected using direct transfer (Van der Plas et al., 2011), i.e. directly following word alignments. Annotations that span several words, such as semantic roles, are projected onto aligned target language constituents that are identified following the procedure introduced in Padó and Lapata (2009). Next to the above-illustrated method call which transfers all linguistic annotation, users may also choose only a subset of annotation types to be projected. For instance, a user may only be interested in projecting named entities, while using existing target language parsers to identify syntactic structure.

Annotation projection produces a target language SENTENCE object with added linguistic annotation. This object can then be saved in CoNLL-U format, for instance for verification or for use in training of target language parsers. Refer to Figure 2 for an illustration.

3.5. Visualization

As discussed above, previous works found that the viability of annotation projection depends both on the type of annotation being projected as well as the specific language pair in question. For this reason, previous work often conducted careful qualitative analyses of error sources and defined strategies to address these issues. To assist such analysis, we previously presented a demonstration of a web-based UI – called THE PROJECTOR (Akbik and Vollgraf, 2017) – that visualizes sentence pairs, word alignments and various layers of linguistic annotation. We include the visualization capabilities of THE PROJECTOR into ZAP.

To launch the interactive UI, a simple method call suffices:

```
int port = 9000;
TheProjectorUI.instance()
    .startServerAtPort(port);
```

This launches the web UI locally at the specified port. For more information on the functionality of THE PROJECTOR, refer to Akbik and Vollgraf (2017). A screenshot of the

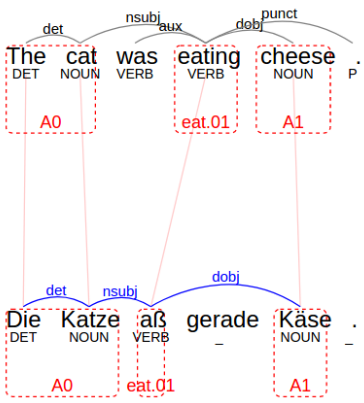

Figure 3: ZAP's visualization of the running example.

example sentence pair as rendered in the UI is illustrated in Figure 3.

4. Summary and Outlook: Extending ZAP

To facilitate experimentation with different strategies, we designed simple interfaces for ZAP that allow researchers to extend the framework. It is our hope that the open source nature of the project will encourage more research into annotation projection, eventually leading to automatically generated linguistic resources that approach the quality of expert annotation. Our current work focuses on extending the framework to support more languages and types of annotation, as well as adding heuristic methods for addressing translational divergences.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732328 ("FashionBrain").

5. Bibliographical References

Akbik, A. and Vollgraf, R. (2017). The projector: An interactive annotation projection visualization tool. In *EMNLP 2017, Conference on Empirical Methods on Natural Language Processing*, pages 43–48.

Akbik, A., Chiticariu, L., Danilevsky, M., Li, Y., Vaithyanathan, S., and Zhu, H. (2015). Generating high quality proposition banks for multilingual semantic role labeling. In *ACL 2015, 53rd Annual Meeting of the Association for Computational Linguistics*, pages 397–407.

Bentivogli, L. and Pianta, E. (2005). Exploiting parallel texts in the creation of multilingual semantically annotated resources: the multisemcor corpus. *Natural Language Engineering*, 11(3):247–261.

Björkelund, A., Hafdell, L., and Nugues, P. (2009). Multilingual semantic role labeling. In *CoNLL 2009, 13th Conference on Computational Natural Language Learning: Shared Task*, pages 43–48.

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *COLING 2010, 23rd International Conference on Computational Linguistics*, pages 89–97.

DeNero, J. and Liang, P. (2007). The berkeley aligner. `http://code.google.com/p/berkeleyaligner/`.

Dorr, B. J. (1994). Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of ibm model 2. In *NAACL 2013, North American Chapter of the Association for Computational Linguistics*, pages 644–648.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL 2014, 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Mayhew, S., Tsai, C.-T., and Roth, D. (2017). Cheap translation for cross-lingual named entity recognition. In *EMNLP 2017, Conference on Empirical Methods in Natural Language Processing*, pages 2526–2535.

Ni, J., Dinu, G., and Florian, R. (2017). Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. *CoRR*, abs/1707.02483.

Padó, S. and Lapata, M. (2009). Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36(1):307–340.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Täckström, O., McDonald, R., and Uszkoreit, J. (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *NAACL 2012, North American Chapter of the Association for Computational Linguistics*, pages 477–487.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *LREC 2012, 8th International Conference on Language Resources and Evaluation*, pages 2214–2218.

Van der Plas, L., Merlo, P., and Henderson, J. (2011). Scaling up automatic cross-lingual semantic role annotation. In *ACL 2011, 49th Annual Meeting of the Association for Computational Linguistics*, pages 299–304.

Wang, C., Akbik, A., Chiticariu, L., Li, Y., Xia, F., and Xu, A. (2017). CROWD-IN-THE-LOOP: A hybrid approach for annotating semantic roles. In *EMNLP 2017, Conference on Empirical Methods in Natural Language Processing*, pages 1914–1923.

Yarowsky, D. and Ngai, G. (2001). Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *NAACL 2001, North American Chapter of the Association for Computational Linguistics*, pages 1–8.

Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *HLT 2001, 1st International Conference on Human Language Technology Research*, pages 1–8.