Multi-Level Alignments As An Extensible Representation Basis for Textual Entailment Algorithms

Tae-Gil Noh (noh@cl.uni-heidelberg.de)[1], **Sebastian Padó** (pado@ims.uni-stuttgart.de)[2], **Vered Shwartz** (vered1986@gmail.com)[3], **Ido Dagan** (dagan@cs.biu.ac.il)[3], **Vivi Nastase** (nastase@fbk.eu)[4], **Kathrin Eichler** (kathrin.eichler@dfki.de)[5], **Lili Kotlerman** (lili.dav@gmail.com)[3], and **Meni Adler** (meni.adler@gmail.com)[3]

[1]Institute of Computational Linguistics, Heidelberg University, Germany
[2]Institute of Natural Language Processing, Stuttgart University, Germany
[3]Department of Computer Science, Bar-Ilan University, Israel
[4]Human Language Technology, Fondazione Bruno Kessler, Italy
[5]Language Technology Lab, DFKI GmbH, Germany

Abstract

A major problem in research on Textual Entailment (TE) is the high implementation effort for TE systems. Recently, interoperable standards for annotation and preprocessing have been proposed. In contrast, the algorithmic level remains unstandardized, which makes component re-use in this area very difficult in practice. In this paper, we introduce *multi-level alignments* as a central, powerful representation for TE algorithms that encourages modular, reusable, multilingual algorithm development. We demonstrate that a pilot open-source implementation of multi-level alignment with minimal features competes with state-of-the-art open-source TE engines in three languages.

1 Introduction

A key challenge of Natural Language Processing is to determine what conclusions can be drawn from a natural language text, a task known as *Textual Entailment* (TE, Dagan and Glickman 2004). The ability to recognize TE helps dealing with surface variability in tasks like Question Answering (Harabagiu and Hickl, 2006), Intelligent Tutoring (Nielsen et al., 2009), or Text Exploration (Berant et al., 2012). Open source implementations a number of TE algorithms have become available over the last years, including BIUTEE (Stern and Dagan, 2012) and EDITS (Kouylekov and Negri, 2010), which has made it much easier for end users to utilize TE engines.

At the same time, the situation is still more difficult for researchers and developers. Even though recently a common platform for TE has been proposed (Padó et al., 2015) that standardizes important aspects like annotation types, preprocessing, and knowledge resources, it largely ignores the algorithmic level. In fact, TE algorithms themselves are generally not designed to be extensible or interoperable. Therefore, changes to the algorithms – like adding support for a new language or for new analysis aspect – are often very involved, if not impossible. This often forces the next generation of TE researchers to develop and implement their own core algorithms from scratch.

In this paper, we address this problem by proposing a schema for TE algorithms that revolves around a central representation layer called *multi-level alignment* geared towards encoding the relevant information for deciding entailment. The use of multi-level alignments encourages a modular, extensible development of TE algorithms that can be partitioned into "alignment producers" and "alignment consumers". This enables for future researchers and developers to change analysis components or add new ones in a straightforward manner.

We also present evaluation results for a very simple TE algorithm based on multi-level alignments for English, German and Italian. It utilizes a minimal set of analyzers and four basic language-independent features. It can thus be regarded as a baseline of the performance achievable with this approach. The results can already compete with the best open-source engines available for each of the languages.

2 TE with Multi-Level Alignments

The quality of the word alignment between a Text (T) and a Hypothesis (H) has been used very early as a simple feature to decide about TE. When it was found

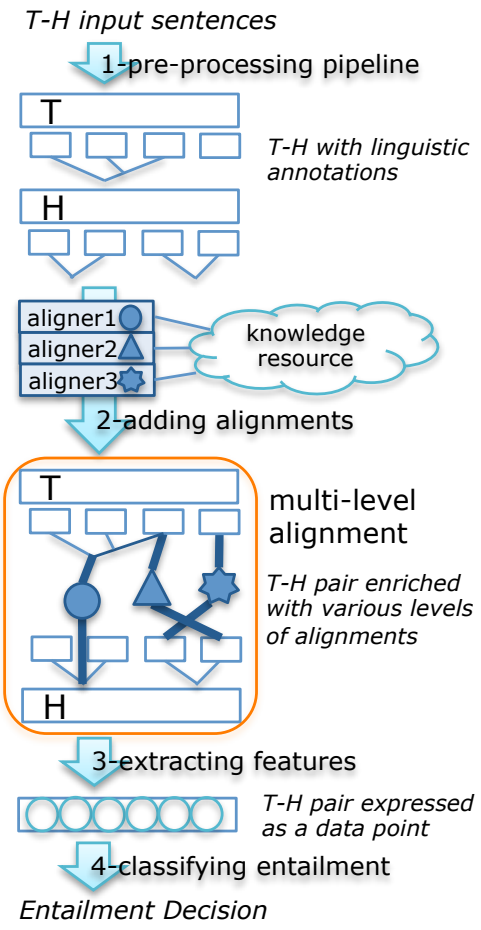

Figure 1: Dataflow for TE algorithms based on multi level alignment

that alignment strength can be misleading (MacCartney et al., 2006), alignment was understood as an intermediate step whose outcome is a set of correspondences between parts of T and H that can be used to define (mis-)match features. Alignments can be established at the word level, phrase level (MacCartney et al., 2008), or dependency level (Dinu and Wang, 2009). Dagan et al. (2013) generalized this practical use to an architectural principle: They showed that various TE algorithms can be mapped onto a universal alignment-based schema with six steps: preprocessing, enrichment, candidate alignment generation, alignment selection, and classification.

Proposal. Our proposal is similar to, but simpler than, Dagan et al.'s. Figure 1 shows the data flow.

First, the text and the hypothesis are linguistically pre-processed. Then, the annotated T-H pair becomes the input for various independent aligners, which have access to knowledge resources and can compute any evidence for or against entailment that can be represented as a weighted alignment between any linguistic levels of H and T. Note that this includes many analyses not normally treated as alignment, e.g. match or mismatch in negation or modality between parts of T and H. The union of all alignments forms the central data structure, the *Multi-Level Alignments*.

The next step is feature extraction. Features can be extracted on the basis of individual alignments, or from sets of alignments. We assume that the features form a vector describing the T-H pair, and that the last step is supervised entailment classification.

Discussion. The main difference to Dagan et al.'s schema is that we intentionally leave out the step of *alignment selection* which explicitly selects a single alignment for each part of H or T, typically the globally most probable one. Our decision to forgo selection is grounded in our design of multi-level alignments as a repository that supports coexistence of information from different sources. This has the following benefits: (a) aligners become decoupled in that adding a new aligner does not have a direct impact on other aligners; (b) alignments produced by different aligners can have different semantics, e.g. positive (match) or negative (mismatch); (c) interactions between alignments can still be captured by defining features in the feature extraction step.

In this manner, multi-level alignments serve as an abstraction layer that encourages the development of TE algorithms composed of small, self-contained modules that solve specialized tasks in TE recognition. Each of these modules consists of two parts: an aligner, and a set of feature extractors. A priori, each module can be defined independently; to introduce interactions with other modules, it should be sufficient to extend the feature extractors.

The practical benefit for the developer is that even relatively complex TE algorithms use a small set of well-defined interfaces, which makes them easy to manage, even at the implementation level. The startup cost is getting acquainted with the common data structure of multi-level alignments. We believe that developers are willing to pay this cost, especially when this provides them with a platform that supports multilingual pre-processing and resources.

3 Implementation and Evaluation

We describe an implementation of a pilot TE algorithm based on the Multi-Level Alignment approach and its evaluation in three languages (EN, DE, IT). The system is available as open-source.[1]

3.1 Technical Foundations

We implement the algorithm within an open source TE development platform (Padó et al., 2015). The platform provides various multilingual pre-processing pipelines and knowledge resources such as WordNet, VerbOcean, etc., under a shared API. For pre-processing, we use TreeTagger-based pipelines for all three languages.

Another important service provided by the platform is the ability of storing a wide range of linguistic annotations in a common, language-independent data representation. The platform uses UIMA CAS (Ferrucci and Lally, 2004) as the data container, adopts the DKPro type system (de Castilho and Gurevych, 2014), and defines annotation types which can be extended in a controlled manner. We used this capability to define a multilingual Multi-Level Alignment layer with little implementation effort.

3.2 A Minimal Set of Aligners

The pilot algorithm restricts itself to three aligners. All three are fully language-independent, even if two use language-specific knowledge resources.

Lexical Aligner. The lexical aligner adds an alignment link for a pair of lemmas in T and H if it finds some kind of semantic relationships between them in a set of lexical resources. The link is directed, labeled (by the semantic relation, e.g. "synonym", "antonym") and weighted, with the weight indicating the strength of the relationship. Note that this aligner can on its own already produce alignment links with inconsistent semantics (positive and negative). For English, WordNet and VerbOcean were used as lexical resources. Italian WordNet was used for Italian, and GermaNet and German DerivBase (Zeller et al., 2013) were used as lexical resources for German.

[1] As a part of Excitement Open Platform for Textual Entailment. `https://github.com/hltfbk/EOP-1.2.1/wiki/AlignmentEDAP1`

Paraphrase Aligner. The paraphrase aligner concentrates on surface forms rather than lemmas and can align sequences of them rather than just individual tokens. It uses paraphrase tables, e.g. extracted from parallel corpora (Bannard and Callison-Burch, 2005). The alignment process is similar to the lexical aligner: any two sequences of tokens in T and H are aligned if the pair is listed in the resource. The alignment links created by this aligner instantiate only one relation ("paraphrase") but report the strength of the relation via the translation probability. We used the paraphrase tables provided by the METEOR MT evaluation package (Denkowski and Lavie, 2014), which are available for numerous languages.

Lemma Identity Aligner. This aligner does not use any resources. It simply aligns identical lemmas between T and H and plays an important role in practice to deal with named entities.

3.3 A Minimal Feature Set

Similar to the aligners, we concentrate on a small set of four features in the pilot algorithm. Again, the features are completely language independent, even at the implementation level. This is possible because the linguistic annotations and the alignments, use a language-independent type system (cf. Section 3.1).

All current features measure some form of *coverage* on the Hypothesis, i.e. the percentage of H that can be explained by T. The underlying hypothesis is that a higher coverage of H corresponds to a higher chance of entailment. Since parts-of-speech arguably differ in the importance of being covered, we compute coverage for four sets of words separately: (a), for all words; (b), for content words; (c), for verbs; (d), for proper names (according to the POS tagger). The features are defined on the union of all produced alignments: i.e., two words count as aligned if they were aligned by any aligner. Clearly, this is an overly simplistic (albeit surprisingly effective) strategy. It can be considered a baseline for our approach that can be extended with many features that suggest themselves from the literature.

4 Experimental Evaluation

Evaluation 1: RTE-3. RTE-3 was the third instance of the yearly benchmarking workshops of the Textual Entailment community (Giampiccolo et al.,

	English	German	Italian
MultiAlign	**67.0**	**64.5**	**65.4**
BIUTEE	**67.0**	-	-
TIE	65.2	63.1	-
EDITS	63.6	-	62.6
RTE3 median	61.8		

Table 1: Accuracy evaluation on the RTE3 dataset

	English	German	Italian
MultiAlign	69.2	**72.4**	**69.5**
BIUTEE	**71.3**	-	-
TIE	67.3	**72.4**	-
EDITS	66.6	-	65.6

Table 2: F_1 evaluation on application data

2007). The English dataset created for RTE-3 consists of 800 training and 800 testing T-H pairs. Later, the RTE-3 dataset was translated into both German and Italian (Magnini et al., 2014). It is the only Textual Entailment dataset in multiple languages with the same content. The task is binary TE recognition, with baseline of 50% accuracy (balanced classes).

We trained and tested our Multi-Level Alignment approach (*MultiAlign*) on the RTE-3 dataset separately for each language. We compare against the other RTE systems from the platform by Padó et al. (2015), namely BIUTEE (Stern and Dagan, 2012), EDITS (Kouylekov and Negri, 2010), and TIE (Wang and Zhang, 2009). Each system is configured with its best known configurations. The pilot system supports all three languages, while others support one (BIUTEE) or two languages (EDITS, TIE).

The results are shown in Table 1. The pilot system performs well in all three languages. It ties with BIUTEE on English and it outperforms TIE and EDITS in their respective results on German and Italian. This is particularly notable since all three systems have gone through several years of development, while *MultiAlign* is only a pilot implementation.

Evaluation 2: T-H pairs from Application Data. We perform the second evaluation on real-world application data from two application datasets: an entailment graph dataset (for English and Italian), and an e-mail categorization dataset (for German). Entailment graph building is the task of constructing graphs that hierarchically structure the statements from a collection (Berant et al., 2012) for the application of Text Exploration. In TE-based e-mail categorization, the goal is to assign the right category to an email with TE, using the email as T and a category description as H. (Eichler et al., 2014).

Due to space constraints, we cannot evaluate these applications end-to-end. Instead, we focus on the respective first step, the binary decision of entailment for individual T-H pairs. This task corresponds to RTE-3, and the main difference to Evaluation 1 is that these pairs come from real-world interactions and were produced by native speakers. All T-H pairs are sampled from application gold data which were manually constructed on the basis of anonymized customer interactions (Eichler et al. (2014) for German; Kotlerman et al. (2015) for English and Italian[2]). The sets are fairly large (5300 pairs for English, 1700 for Italian, 1274 for German), and were sampled to be balanced. We report F_1 for comparability with non-balanced setups (our random baseline is F_1=50).

Table 2 shows our evaluation results. *MultiAlign* system beats EDITS for Italian (+4), and ties with TIE for German. On English, BIUTEE still outperforms *MultiAlign* (-2). Thus, *MultiAlign* also performs acceptably on real-world data.

In sum, we find that *MultiAlign* is already competitive with state-of-the-art open-source TE engines on three languages. *MultiAlign* is not only much less complex, but it is also a single system covering all three languages, without any language-specific optimizations. We interpret this as a positive sign for the future of the Multi-Level Alignment approach.

Visualization. The platform also supports visualization of individual Text-Hypothesis pairs, showing the alignments that were created by the system as well as the features computed on the basis of the alignments. The visualization was built on the basis of the BRAT library.[3]

Figure 2 shows an example for the Text *The judges made an assessment of Peter's credibility* and the Hypothesis *The judges assessed if Peter was credible*. The top line shows the final prediction, Entailment, and the confidence (75%). The main part shows the Text and the Hypothesis below each other, connected

[2]Both datasets are publicly available.

[3]`http://brat.nlplab.org/index.html`

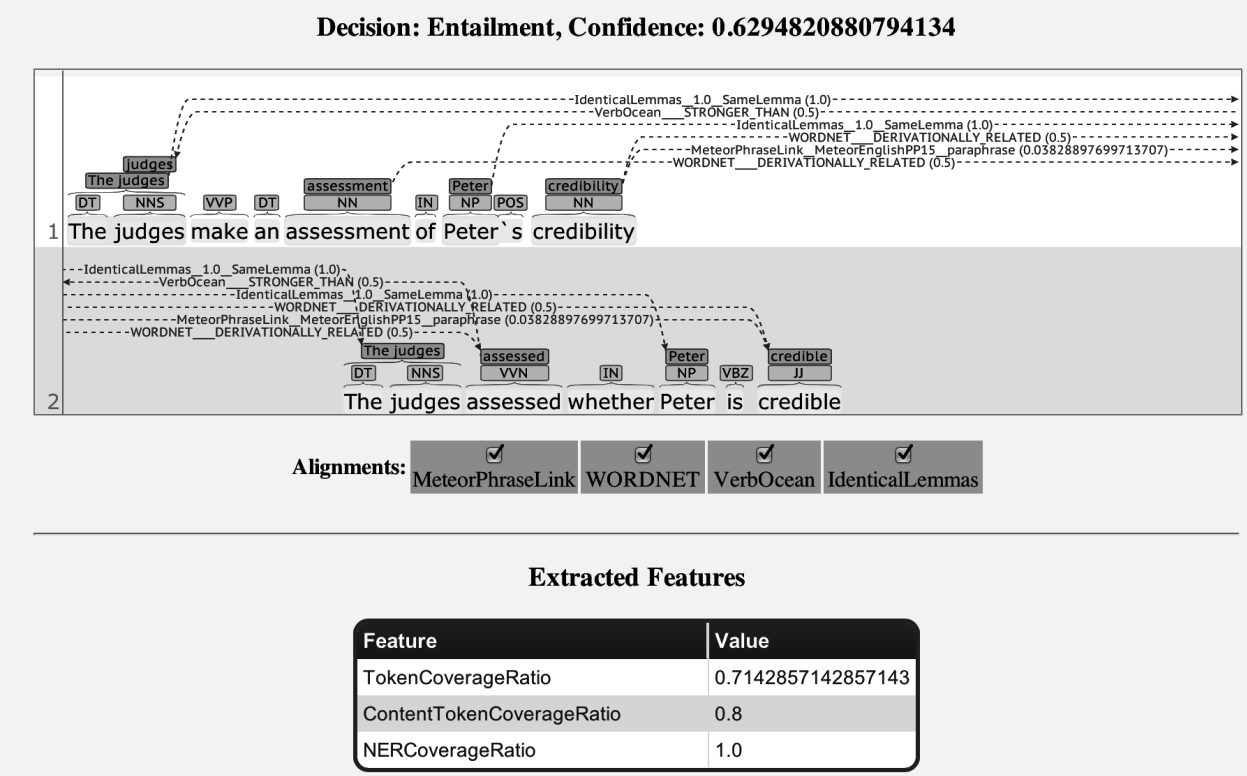

Figure 2: Screenshot of the Multi-Level Alignment Visualizer

by alignment links that are labeled with their source and their score. Recall that we use WordNet and VerbOcean as knowledge sources for the Lexical Aligner, Meteor for the Paraphrase Aligner, and finally the Identical Lemma Aligner. Note that the alignments can link individual words (*assessment* and *assess* are aligned through a derivational link from WordNet) but also phrases (The two occurrences of *The judges* in Text and Hypothesis are linked by virtue of being identical lemmas).

The three features currently used by the English system are are shown below. As can be seen, they aggregate very simple statistics about the alignments: 5 of 7 tokens in the hypothesis are covered, 4 out of 5 content words, and the one proper name is also aligned. This situation motivates nicely the use of those features: a relatively low alignment coverage on all tokens is still compatible with entailment as long as the crucial tokens are aligned.

This visualization enables end users to quickly take in the justification behind the system's decision. Developers can inspect alignments and features for plausibility and detect possible bugs and assess the limitations of aligners and their underlying resources. For example, the current example shows a wrong link produced by the VerbOcean resource between the noun *judges* in the Text and the verb *assessed* in the Hypothesis. The reason is that the noun *judges* is mistaken for an inflected form of the verb *to judge* which indeed stands in a *Stronger-than* relationship to *to assess*.

5 Conclusion

This paper proposed the use of *multi-level alignments*, a rich data structure allowing multiple alignments to co-exist. We argued that multi-level alignments are a suitable basis for developing Textual Entailment algorithms by virtue of providing a beneficial abstraction layer that supports extensible and modular entailment algorithms. A pilot TE algorithm developed in this schema showed performance comparable to much more sophisticated state-of-the-art open-source TE engines and is available as open source software.

References

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 597–604, Ann Arbor, MI.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2012. Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

Ido Dagan and Oren Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *PASCAL workshop on Learning Methods for Text Understanding and Mining*, Grenoble, France.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. *Recognizing Textual Entailment: Models and Applications*. Number 17 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT at COLING 2014*, pages 1–11, Dublin, Ireland, August.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, MD.

Georgiana Dinu and Rui Wang. 2009. Inference rules and their application to recognizing textual entailment. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 211–219, Athens, Greece.

Kathrin Eichler, Aleksandra Gabryszak, and Günter Neumann. 2014. An analysis of textual inference in German customer emails. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics*, pages 69–74, Dublin, Ireland.

David Ferrucci and Adam Lally. 2004. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3–4):327–348.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognising textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, Czech Republic.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 905–912, Sydney, Australia.

Lili Kotlerman, Ido Dagan, Bernardo Magnini, and Luisa Bentivogli. 2015. Textual entailment graphs. *Natural Language Engineering*.

Milen Kouylekov and Matteo Negri. 2010. An open-source package for recognizing textual entailment. In *Proceedings of the ACL 2010 System Demonstrations*, pages 42–47, Uppsala, Sweden.

Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 41–48, New York City, USA.

Bill MacCartney, Michel Galley, and Christopher D Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 802–811, Honolulu, Hawaii.

Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Günter Neumann, Tae-Gil Noh, Sebastian Padó, Asher Stern, and Omer Levy. 2014. The EXCITEMENT open platform for textual inferences. In *Proceedings of the ACL 2014 System Demonstrations*, pages 43–48, Baltimore, MD.

Rodney D Nielsen, Wayne Ward, and James H Martin. 2009. Recognizing entailment in intelligent tutoring systems. *Journal of Natural Language Engineering*, 15(4):479–501.

Sebastian Padó, Tae-Gil Noh, Asher Stern, Rui Wang, and Roberto Zanoli. 2015. Design and realization of a modular architecture for textual entailment. *Journal of Natural Language Engineering*, 21(2):167–200.

Asher Stern and Ido Dagan. 2012. BIUTEE: A modular open-source system for recognizing textual entailment. In *Proceedings of the ACL 2012 System Demonstrations*, pages 73–78, Jeju Island, Korea.

Rui Wang and Yi Zhang. 2009. Recognizing textual relatedness with predicate-argument structures. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 784–792, Singapore.

Britta Zeller, Jan Šnajder, and Sebastian Padó. 2013. DErivBase: Inducing and evaluating a derivational morphology resource for German. In *Proceedings of ACL*, pages 1201–1211, Sofia, Bulgaria.