

Design and Realization of the EXCITEMENT Open Platform for Textual Entailment Günter Neumann, DFKI Sebastian Pado, Universität Stuttgart

Textual Entailment (TE)

.

 A Text (T) entails a Hypothesis (H), if a typical human reading T would infer that H is most likely true [Dagan et al. 2005]

Logical entailment:

- A formula A entails a formula B if in all models where A holds, B holds as well. [e.g., Chierchia & McConnell-Ginet 2002]
- TE is agnostic with regard to representation of T and H
- TE is defined by human judgments and not model theory
- TE captures "common sense reasoning": Inclusion of almost certain entailments

The promise of Textual Entailment

- Semantic processing is a very fragmented research area
 - Many phenomena
 - Many approaches
 - Many applications
- Can TE be a unifying paradigm for semantic processing?
- Claim: Many NLP tasks can be "powered" by entailment
 - Question Answering: document text must entail answer candidate [e.g., Harabagiu and Hickl 2006]
 - Automatic Tutoring: student answer must entail reference answer [e.g., Nielsen et al. 2009]
 - Information Presentation: show entailment hierarchy [e.g., Berant et al. 2012]



- Textual Entailment was proposed in 2004
 - Since then: Yearly Recognition of Textual Entailment (RTE) shared tasks
- Ten years of research
 - Much progress regarding algorithms, resources, …
- Three main groups of algorithms:
 - Alignment-based: Align words in Text and Hypothesis
 - Transformation-based: Rewrite Text into Hypothesis
 - Formal language-based: Represent Text and Hypothesis in formal language and apply reasoning methods



- Many research prototype system:
- Two open source systems for Textual Entailment:
 - EDITS, an alignment-based system (FBK)
 - http://edits.fbk.eu
 - BIUTEE, a transformation-based system (BIU)
 - http://u.cs.biu.ac.il/~nlp/downloads/biutee/protected-biutee.html

- Does this mean that TE technology is easy to use and understand?
- No, we are not there yet

- Systems are prototypes of specific algorithms
 - Hard-wired preprocessing tools
 - Hard-wired assumptions about language
 - No modularization of algorithmic parts
 - No interchange format for inference rules





- Research project funded by European Commission (FP 7)
 Academic Partners: BIU, DFKI, FBK, HEI
- Goal: Infrastructure for sustainable research in TE
- EXCITEMENT Open Platform (EOP): A TE suite that is
 - Multilingual
 - Component-based
 - Open source

Specification: Modular architecture for TE systems

Reusability of algorithms, resources through interfaces

. . .

- Towards "plug and play" construction of systems
- Platform: Implementation of modular specification
 - Multilingual: TE systems for English, German, Italian

- Both complete in first releases
- This presentation: Highlights
 - More details in the tutorial this afternoon



The EOP specification

The EOP Architecture



EXCITEMENT

Linguistic Analysis Pipeline

n a na fan a fan in ar

- Apache UIMA: linguistic analysis = enrichment of document with strongly typed annotation
- DKPro type system: language-independent representation of (almost) all linguistic layers [Gurevych et al. 2007]
- Entailment Core (Java-based)
 - Interfaces for relevant modules
- Some glue
 - E.g., common configuration
- Also: "soft" constraints ("best practice" policies)
 - Initialization behavior, error handling, ...

a de la section de

- Top-level interface: Entailment Decision Algorithm
 - Text-Hypothesis pair (UIMA) in, Decision out
 - Existing systems can be wrapped trivially as EDAs
- Three major component types
 - Annotation components
 - Feature components
 - Knowledge components

the second s

Components

Annotation components

 Add linguistic analysis to the P/H pair, e.g. alignment



- Feature components
 - Compute match/mismatch features, distance/similarity features, scoring features, ...
- Knowledge components
 - Provide access to inference rule bases
 - □ Lexical inference rule: Lemma₁ → Lemma₂ Dog → animal, snore → sleep
 - □ Lexical-syntactic inference rule: Tree fragment₁ → Tree fragment₂ X buy Y from Z → X pays Z for Y













The EOP implementation



First release of EOP is available for download!

- GPL licensed
- EDAs
 - Three EDAs, EDITS, TIE, and BIUTEE
- LAPs
 - For three languages
- Datasets (Based on RTE-3 data)
 - English, German, Italian, 1600 T-H pairs for each
- Various components and many knowledge resources
- Documentation and Tutorials

http://hltfbk.github.io/Excitement-Open-Platform/



EXCITEMENT-Open-Platform

for Textual Entailment

/// Introduction

The EXCITEMENT Open Platform (EOP) is an open source software platform containing stateof-the-art algorithms for recognizing texual entailment relations:

given two text fragments, one named text and the other named hypothesis, the task consists in recognizing whether the hypothesis can be inferred from the text.

Written in Java, EOP is a main product of the project EXCITEMENT - EXploring Customer Interactions through Textual EntailMENT which is funded by the European Commission under the European Union's Seventh Framework Programme (FP7). EOP is designed to be efficient and extendable. Highlights include:

- · Separation between Linguistic Analysis Pipelines and Entailment Components
- · Supporting modularity and interoperability among Components
- · Java API with source code
- Pre-trained multi-lingual models (i.e. English, German, Italian)
- · Trainable with new sample data
- Detailed documentation to understand the structure and implementation of EOP
- Quick Start documentation to start using EOP from now.

Reference: S. Pado, T.-G. Noh, A. Stern, R. Wang, R. Zanoli: Design and Realization of a Modular Architecture for Textual Entailment. To appear in Natural Language Engineering. Copyright Cambridge University Press, 2013.

/// Licence

The EXCITEMENT Open Platform (EOP) is released under the terms of General Public License (GPL) version 3.

/// Code Distribution

We provide different distributions for users depending on the fact that they want to get the code as a Zip File, clone the code from the EOP GitHub repository or they want to use the EOP Maven Artifacts. Users have access to the lastest code release by selecting the Lastest Release option at the right side



November 20-22, 2013 - The Joint Symposium on Semantic Processing (JSSP 2013) will take place at FBK in Trento (Italy).

AUGUST 2013: Release v1.0.2 available.





TIME

When the source code in the master branch reaches a stable point, all of the changes are merged back into a release, and are tagged with a release number.

Jenkins, the continuous integration tool



Jenkins monitors both the master and the release branch in the EOP GitHub repository, and whenever it detects a commit to a branch, it builds and tests the code in the branch.

a sa si sa si sa sa s



EXCITEMENT

Beta testers

- Test the EOP by performing some benchmark
- E.g., Vo Ngoc Phuoc An (FBK) on RTE-2 data sets

Users

- Use EOP as part of a project, mainly as a black box
- E.g., Inside Excitement (Transduction layer), BMBF-funded project MEDIXIN (DFKI), HEI fall school (CL students), starting Master/PhD student projects (DFKI, FBK)

Developers

- Contribute extensions to the EOP
- E.g., PhD project by Daniel Bär (UKP-Lab, TU Darmstadt)

- Users: EOP works, but is still difficult to install and use
 - Lack of documentation: Ongoing tutorial development

. . . .

- Inherent complexity of setup: Packaging EOP into VM
- EOP is used inside and outside EXCITEMENT
 - As part of Excitement: Entailment graph, IR query expansion, application of EDITS in HEI to social media data
 - As part of external partners: Entailment-based QA
- 2nd cycle of EOP specification until Spring 2014
 - Addressing shortcomings of the first specification
 - Extending the specification to include logic-based TE systems (Beltagy et al. 2013)

Take full advantage of the EOP's "toolbox" architecture

- Use as evaluation platform for systems or knowledge on RTE data
- E.g., influence of phrase similarity from distributional models of similarity on Textual Entailment

- Turn EOP into a fully open source project
 - Project EXCITEMENT runs until 12/2014

and the second second

- Gradually release control to open source community
- Model: MOSES



- EXCITEMENT web site: http://www.excitement-project.org
 - Specification document
- S. Pado, T-G. Noh, A. Stern, R. Wang, R. Zanoli: Design and Realization of a Modular Architecture for Textual Entailment. Accepted for publication in *Natural Language Engineering*. Preprints available from the authors' pages.
- T.-G. Noh, S. Pado. Using UIMA to structure an Open Platform for Textual Entailment. 2013. Proceedings of the UIMA@GSCL workshop.