# Entailment Graphs for Text Analytics in the Excitement Project

Bernardo Magnini[1], Ido Dagan[2], Günter Neumann[3], and Sebastian Pado[4]

[1] Fondazione Bruno Kessler, Trento, Italy
magnini@fbk.eu
[2] Bar Ilan University, Israel
dagan@cs.biu.ac.il
[3] DFKI, Germany
neumann@dfki.de
[4] Stuttgart University
pado@ims.uni-stuttgart.de

**Abstract.** In the last years, a relevant research line in Natural Language Processing has focused on detecting semantic relations among portions of text, including entailment, similarity, temporal relations, and, with a less degree, causality. The attention on such semantic relations has raised the demand to move towards more informative meaning representations, which express properties of concepts and relations among them. This demand triggered research on "statement entailment graphs", where nodes are natural language statements (propositions), comprising of predicates with their arguments and modifiers, while edges represent entailment relations between nodes.

We report initial research that defines the properties of entailment graphs and their potential applications. Particularly, we show how entailment graphs are profitably used in the context of the European project EXCITEMENT, where they are applied for the analysis of customer interactions across multiple channels, including speech, email, chat and social media, and multiple languages (English, German, Italian).

**Keywords:** Semantic inferences, textual entailment, text analytics.

## 1 Introduction

Textual entailment [4] suggests a long-term research direction where language understanding can take advantage from the capacity to resolve text-to-text semantic inferences. Among the others, the community has focused on two related aspects: the Recognizing Textual Entailment (RTE) shared task [3], aimed at capturing entailment between two portions of text, and knowledge acquisition, which aims at large-scale acquisition of entailment rules. On the application side, text-to-text semantic inferences may help in several scenarios where content analytics needs to be exploited. In this area taxonomy-based representations are currently widely used to model compactly large amounts of textual data. However, while current methods allow organizing knowledge at the lexical level (keywords/concepts/topics), there is an increasing demand to move towards more informative representations, which express properties of concepts and relations

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 11–18, 2014.

among them. This demand triggered our research on statement entailment graphs. In these graphs, nodes are natural language statements (propositions), comprising of predicates with their arguments and modifiers, while edges represent entailment relations between nodes. In this paper we report initial research that defines the properties of entailment graphs and their potential applications. Particularly, we show how entailment graphs can be profitably used for both knowledge acquisition and text exploration. Beyond providing a rich and informative representation, statement entailment graphs allow integrating multiple semantic inferences. So far, textual inference research focused on single, mutually independent, entailment judgments. However, in many scenarios there are dependencies among Text/Hypothesis pairs, which need to be captured consistently. This calls for global optimization algorithms for inter-dependent entailment judgments, taking advantage of the overall entailment graph structure (e.g. ensuring entailment graph transitivity).

From the applied perspective, we are experimenting with entailment graphs in the context of the EXCITEMENT project[1] industrial scenarios. We focus on the text analytics domain, and particularly on the analysis of customer interactions across multiple channels, including speech, email, chat and social media, and multiple languages (English, German, Italian). For example, we would like to recognize that the complaint *"they charge too much for sandwiches"* entails *"food is too expensive"*, and allow an analyst to compactly navigate through an entailment graph that consolidates the information structure of a large number of customer statements. Our eventual applied goal is to develop a new generation of inference-based text exploration applications, which will enable businesses to better analyze their diverse and often unpredicted client content.

The paper is structured as follows. Section 2 introduces the general aspects of entailment graphs as a tool for text analytics. Section 3 presents the architecture of the platform for textual inferences, which is at the core of the construction of entailment graphs. Finally, Section 4 provides an overview of the use and of the advantages of entailment graphs applied for customer interaction analysis.

## 2 Entailment Graphs

Recently, entailment graphs have been proposed ([1], [2]) as an efficient and informative organization of entailment rules automatically acquired from corpora. In this context, a node in a entailment graph is supposed to represent a simple statement composed by a predicate with its (possibly typed) arguments, while direct edges among nodes indicate an entailment relation. As an example, from [1], a node like "X-reduce-nausea" entails a node like "X-help-with-nausea", assuming the same instantiation for the X variable. In a entailment graph nodes are natural language statements (propositions), comprising of predicates with their arguments and modifiers, while edges represent entailment relations between nodes. Additionally, given that entailment is a directed and transitive relation, a well formed entailment graph should preserve transitivity among connected nodes.

In the EXCITEMENT project we consider entailment graphs based on more complex statements, where, in addition to a single predicate, we include grammatical modifiers

[1] `http://www.excitement-project.eu`

both of the predicate and of its arguments. For example, the sentence *"Lights in night trains are annoyingly bright"*, expresses a predication about the lights in a particular kind of trains. Within a statement we individuate the following basic elements: a top level predicate, usually corresponding to the root node in a dependency tree, expressed by the word "bright" in the example, one or more arguments of the predicate (i.e. the word "lights"), and a number of grammatical modifiers, either of the predicate (i.e. "annoyingly") or of the arguments of the predicate (i.e. "in night trains").

Grammatical modifiers, i.e. tokens which can be removed from the statement without affecting its comprehension, are represented as dependencies of the modifier from other tokens. As an example, given the following output of the Stanford Dependency Parser [6] for our statement:

```
nsubj(bright-7, Lights-1)
nn(trains-4, night-3)
prep_in(Lights-1, trains-4)
cop(bright-7, are-5)
advmod(bright-7, annoyingly-6)
root(ROOT-0, bright-7)
```

we can derive the following head-modifier dependencies: *trains* depends on *lights*, *night* on *trains* and *annoyingly* on *bright*, and the preposition-object dependencies *in-2* on *trains-4* and *trains-4* on *in-2*. Assuming that each head-modifier dependency indicates an entailment relation between the statement with the modifier (e.g. "Lights in night trains are annoyingly bright") and the statement without the modifier (e.g. "Lights are annoyingly bright"), we can recursively apply the procedure to build an entailment graph for the initial statement. The resulting graph is a Directed Acyclic Graph (DAG) generated over the partial order relation determined by the set inclusion over the modifiers in the statement. Figure 2 shows an example of entailment graph. In fact, the resulting DAG for a single statement is a lattice, with the maximal and the minimal statements being respectively the top and the bottom nodes [9], and where the entailment relations are ordered chains in the lattice.

While the construction of the entailment graph for a single statement requires syntactic knowledge (i.e. dependency relations), a more complex situation occurs when it is necessary to merge two entailment graphs, where a broader range of knowledge is necessary to compare two statements. This is addressed exploiting the potential of a text-to-text inference engine, described in the next Section.

## 3 The Excitement Open Platform (EOP)

A major result of the project is the release of the EXCITEMENT Open Platform (EOP). The goal of the platform is to provide functionality for the automatic identification of entailment relations among texts. The EOP is based on a modular architecture with a particular focus on *language-independent* algorithms. It allows developers and users to combine linguistic pipelines, entailment algorithms and linguistic resources within and across languages with as little effort as possible. The result is an ideal software environment for experimenting and testing innovative approaches for textual inferences.

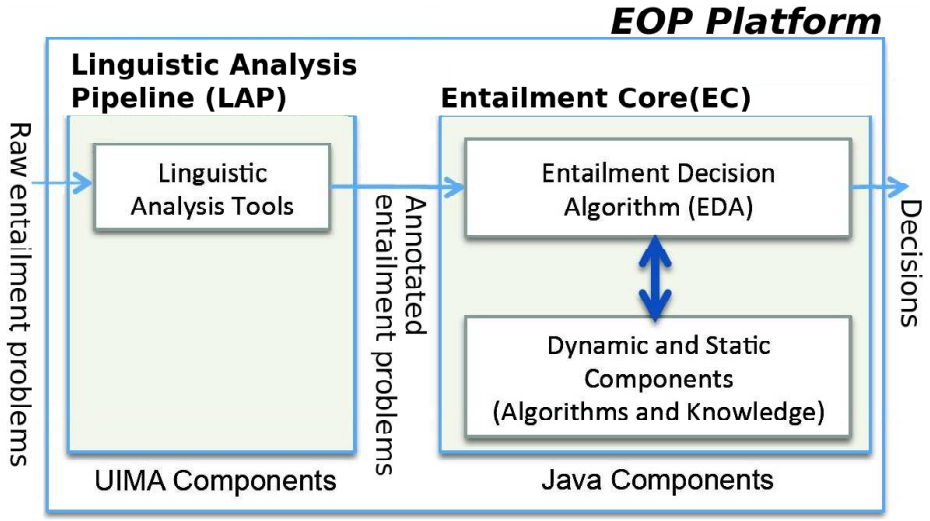


**Fig. 1.** EOP Architecture

The platform is distributed as an open source software[2] and its use is open both to users interested in using inference in applications and to developers willing to extend the current functionalities.

The EOP platform takes as input two text portions, the first called the Text (abbreviated with T), the second called the Hypothesis (abbreviated with H). The output is an entailment judgment, either "Entailment" if T entails H, or "NonEntailment" if the relation does not hold. A confidence score for the decision is also returned in both cases. The EOP architecture ([8], [7]) is based on the concept of modularization with pluggable and replaceable components to enable extension and customization. The overall structure is shown in Figure 1 and consists of two main parts. The Linguistic Analysis Pipeline (LAP) is a series of linguistic annotation components. The Entailment Core (EC) performs the actual entailment recognition. This separation ensures that (a) the components in the EC only rely on linguistic analysis in well-defined ways and (b) the LAP and EC can be run independently of each other. Configuration files are the principal means of configuring the EOP.

The Linguistic Analysis Pipeline is a collection of annotation components for Natural Language Processing (NLP) based on the Apache UIMA framework.[3] Annotations range from tokenization to part of speech tagging, chunking, Named Entity Recognition and parsing. The Entailment Core performs the actual entailment recognition based on the preprocessed text made by the Linguistic Analysis Pipeline. It consists of one or more Entailment Decision Algorithms (EDAs) and zero or more subordinate components. An EDA takes an entailment decision (i.e., "entailment" or "no entailment") while components provide static and dynamic information for the EDA. Scoring

[2] `http://hltfbk.github.io/Excitement-Open-Platform/`
[3] `http://uima.apache.org/`

Components accept a Text/Hypothesis pair as an input, and return a vector of scores. Their output can be used directly to build minimal classifier-based EDAs forming complete RTE systems.

The current version of the EOP platform ships with three EDAs corresponding to three different approaches to RTE: an EDA based on transformations between T and H, an EDA based on edit distance algorithms, and a classification based EDA using features extracted from T and H. Knowledge resources are crucial to recognize cases where T and H use different textual expressions (words, phrases) while preserving entailment. The EOP platform includes a wide range of knowledge resources, including lexical and syntactic resources, where some of them are grabbed from manual resources, like dictionaries, while others are learned automatically. Many EOP resources are inherited from pre-existing RTE systems migrated into the EOP platform, but now use the same interfaces, which makes them accessible in a uniform fashion.

Finally, the EOP infrastructure follows state-of-the-art software engineering standards to support both users and developers. In addition to communication channels, (e.g. mailing list, issue tracking, web site), the platform comprises a version control system, a rich documentation, an archive for storing results, and a package for continuous integration.

## 4 Analysing Customer Interactions

This Section provides details on the EXCITEMENT application scenario as well as how we are manually annotating datasets both for training and for evaluation.

Data are based on real customer interactions and business scenarios with high potential impact for the industrial partners of the project. The datasets cover three languages (English, German and Italian) and three communication channels (speech, email, and social media). All data comply with European and national privacy regulations and will be publicly distributed for research purposes under a Creative Commons license Attribution-NonCommercial-ShareAlike. Two different types of datasets - corresponding to the two main use cases addressed in the project were created performing different kinds of annotation. Section 4.1 describes the novel graph-based annotation, aimed at producing entailment graphs to be used for evaluation within Use Case 1 (text exploration), while Section 4.2 presents more traditional RTE-style entailment datasets, created to test entailment systems within Use Case 2 (information access).

The collected data (some hundreds of interactions for English and Italian) were anonymized where necessary, depending on the partners or customers restrictions. The following is an example (reported as it is, including orthographic and grammatical mistakes) of an anonymized interaction in English, where reasons for dissatisfaction in train service are reported:

> I would not recommend Quasigo as the the tickets are inflexible, I had to change at Moonport instead of Belville europe on the return journey, the food is terribly expensive and not good, there was a fight for luggage space, and on the return journey I did not get the table seats I had booked.

### 4.1 Use Case 1: Text Exploration

To create this dataset, we performed a novel graph-based annotation, which aims to build an entailment graph for a certain number of customer interactions pertaining to a given topic. A customer interaction can be a telephone call, a feedback received by email, a post on a social channel, while the topic gives a general reason for the interaction and can be a business event (e.g. express dissatisfaction, report billing problems, etc.) or any business case that the final user would like to explore.

The entailment graph creation starts from the customer interactions collected for a given topic and includes the following main steps:

- For each interaction, relevant text fragments are extracted (one or more fragment); each fragment contains a specific reason for complaining and corresponds to a single statement (see Section 2);
- For each fragment, a number of subfragments are extracted (see Section 2) and the corresponding entailment graph is manually created;
- All the fragment graphs are merged into the final entailment graph on the base of entailment judgments among statements provided by annotators.

Figure 2 shows an example of entailment graph. Statements within the same node are considered as equivalent (i.e. paraphrases); numbers on nodes indicate the frequency with which a certain statement occurs, while numbers on edges indicate the confidence of the entailment judgment. It is worth to notice that the statement-based representation provided by entailment graphs is much more informative (albeit still very compact) than labels currently provided by document categorization technology, resulting in a more effective tool for text analytics.

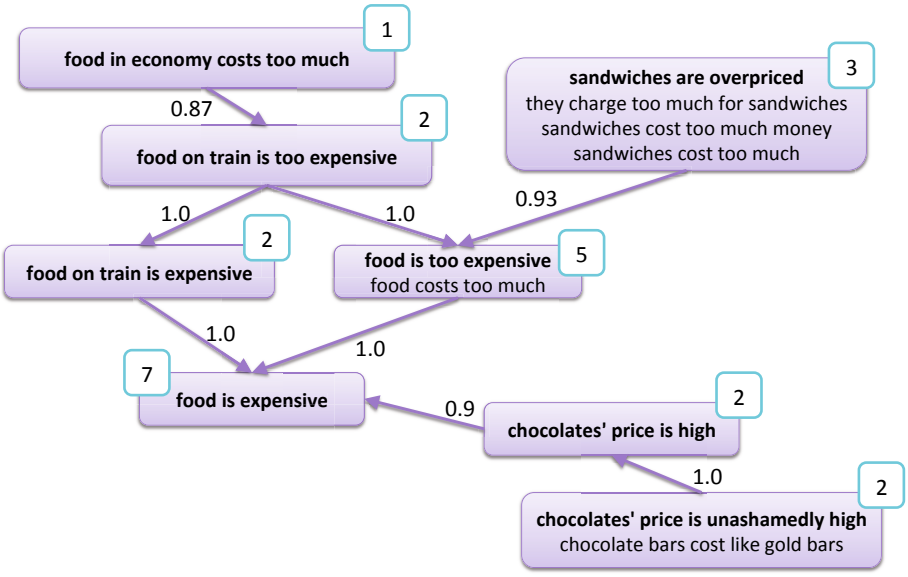


**Fig. 2.** Example of entailment graph for the text exploration use case

### 4.2 Use Case 2: Information Access

Use case 2 is based on a system used by agents in a support center for handling customers inquiries in email format. Customers issues are stored in a database along with their solutions after their handling. When a new email is received, the system maps the problems mentioned in the email with problems stored in the database and thus to automatically suggest a suitable solution to the agent. The mapping is done by ranking the relevance of each stored issue according to presence of keywords characteristic to the issue. Following this ranking process, the system displays the top ranked problems to the agent. In this scenario, the entailment engine (see Section 3 on the EOP) provides a more precise and effective statement-based ranking compared to the keyword-based ranking. Accordingly, a dataset is created following the RTE-style entailment annotation, where stand-alone sentence pairs composed of a text and a hypothesis are created and annotated with the corresponding entailment judgment. This annotation is used to evaluate the entailment engines in isolation, as well as the performance of the Use Case 2 about entailment-driven search at the statement level.

## 5 Conclusion

We have presented the main research lines that are being carried on in the context of the EXCITEMENT project. We are designing an innovative framework for a compact and effective representation of entailment relations among statements based on entailment graphs. We have shown how text-to-text entailment judgments are the core relations for building entailment graphs and we have presented the EOP, an open source platform that provides such text-to-text judgments. We are experimenting the use of entailment graphs and of the EOP for customer interaction analytics, where the relevant content of streams of interactions is detected and then organized in order to facilitate deeper analysis.

Several activities are still ongoing in the EXCITEMENT project. First, an extensive evaluation of the approach, based on manually annotated datasets, all of them publicly available, which we are using for both training and test purposes. A second research line involves a deep investigation of the strategies for automatically building entailment graphs with the EOP platform, including strategies for producing coherent graphs minimizing possible violations of transitivity of the entailment relation.

**Acknowledgments.** This work was partially supported by the EC-funded project EXCITEMENT (FP7ICT-287923).

## References

1. Berant, J., Dagan, I., Goldberger, J.: Global Learning of Focused Entailment Graphs. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 1220–1229 (2010)
2. Berant, J., Dagan, I., Goldberger, J.: Global Learning of Typed Entailment Rules. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, pp. 610–619 (2011)

3. Dagan, I., Glickman, O., Magnini, B.: The Pascal recognising textual entailment challenge. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 177–190. Springer, Heidelberg (2006)
4. Dagan, I., Dolan, B., Magnini, B., Roth, D.: Recognizing textual entailment: Rational, evaluation and approaches. Journal of Natural Language Engineering 15(04), i–xvii (2009)
5. Eads, P., McKay, B.: An Algorithm for generating subset of fixed size with a strong Minimal Change Property. Information Processing Letters 19, 131–133 (1984)
6. Klein, D., Manning, C.D.: Accurate Unlexicalized Parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423–430 (2003)
7. Magnini, B., Zanoli, R., Dagan, I., Eichler, K., Neumann, G., Noh, T.-G., Padó, S., Stern, A., Levy, O.: The Excitement Open Platform for Textual Inferences. To appear in Proceedings of the 52nd Meeting of the Association for Computational Linguistics, Demo papers (2014)
8. Padó, S., Noh, T.-G., Stern, A., Wang, R., Zanoli, R.: Design and Realization of a Modular Architecture for Textual Entailment. Journal of Natural Language Engineering (2014)
9. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Ferré, S., Rudolph, S. (eds.) ICFCA 2009. LNCS, vol. 5548, pp. 314–339. Springer, Heidelberg (2009)