Generating Entailment Rules from FrameNet

Roni Ben Aharon
Department of Computer Science
Bar-Ilan University
Ramat Gan, Israel
r.ben.aharon@gmail.com

Idan Szpektor
Yahoo! Research
Haifa, Israel
idan@yahoo-inc.com

Ido Dagan
Department of Computer Science
Bar-Ilan University
Ramat Gan, Israel
dagan@cs.biu.ac.il

Abstract

Many NLP tasks need accurate knowledge for semantic inference. To this end, mostly WordNet is utilized. Yet WordNet is limited, especially for inference between predicates. To help filling this gap, we present an algorithm that generates inference rules between predicates from FrameNet. Our experiment shows that the novel resource is effective and complements WordNet in terms of rule coverage.

1 Introduction

Many text understanding applications, such as Question Answering (QA) and Information Extraction (IE), need to infer a target textual meaning from other texts. This need was proposed as a generic semantic inference task under the Textual Entailment (TE) paradigm (Dagan et al., 2006).

A fundamental component in semantic inference is the utilization of knowledge resources. However, a major obstacle to improving semantic inference performance is the lack of such knowledge (Bar-Haim et al., 2006; Giampiccolo et al., 2007). We address one prominent type of inference knowledge known as *entailment rules*, focusing specifically on rules between predicates, such as '*cure X* $\Rightarrow$ *X recover*'.

We aim at highly accurate rule acquisition, for which utilizing manually constructed sources seem appropriate. The most widely used manual resource is WordNet (Fellbaum, 1998). Yet it is incomplete for generating entailment rules between predicates (Section 2.1). Hence, other manual resources should also be targeted.

In this work[1], we explore how FrameNet (Baker et al., 1998) could be effectively used for generating entailment rules between predicates. FrameNet is a manually constructed database based on Frame Semantics. It models the semantic argument structure of predicates in terms of prototypical situations called *frames*.

Prior work utilized FrameNet's argument mapping capabilities but took entailment relations from other resources, namely WordNet. We propose a novel method for generating entailment rules from FrameNet by detecting the entailment relations implied in FrameNet. We utilize FrameNet's annotated sentences and relations between frames to extract both the entailment relations and their argument mappings.

Our analysis shows that the rules generated by our algorithm have a reasonable "per-rule" accuracy of about 70%[2]. We tested the generated rule-set on an entailment testbed derived from an IE benchmark and compared it both to WordNet and to state-of-the-art rule generation from FrameNet. Our experiment shows that our method outperforms prior work. In addition, our rule-set's performance is comparable to WordNet and it is complementary to WordNet when uniting the two resources. Finally, additional analysis shows that our rule-set accuracy is 90% in practical use.

2 Background

2.1 Entailment Rules and their Acquisition

To generate entailment rules, two issues should be addressed: a) identifying the lexical entailment relations between predicates, e.g. '*cure* $\Rightarrow$ *recover*'; b) mapping argument positions, e.g. '*cure X* $\Rightarrow$ *X recover*'. The main approach for generating highly accurate rule-sets is to use manually constructed resources. To this end, most systems mainly utilize WordNet (Fellbaum, 1998), being the most prominent lexical resource with broad coverage of predicates. Furthermore, some of its

[1] The detailed description of our work can be found in (Ben Aharon, 2010).

[2] The rule-set is available at: http://www.cs.biu.ac.il/~nlp/downloads

Proceedings of the ACL 2010 Conference Short Papers, pages 241–246,
Uppsala, Sweden, 11-16 July 2010.

relations capture types of entailment relations, including *synonymy*, *hypernymy*, *morphologically-derived*, *entailment* and *cause*.

Yet, WordNet is limited for entailment rule generation. First, many entailment relations, notably for the WordNet *entailment* and *cause* relation types, are missing, e.g. '*elect* $\Rightarrow$ *vote*'. Furthermore, WordNet does not include argument mapping between related predicates. Thus, only *substitutable* WordNet relations (*synonymy* and *hypernymy*), for which argument positions are preserved, could be used to generate entailment rules. The other *non-substitutable* relations, e.g. *cause* ('*kill* $\Rightarrow$ *die*') and *morphologically-derived* ('*meet.v* $\Leftrightarrow$ *meeting.n*'), cannot be used.

2.2 FrameNet

FrameNet (Baker et al., 1998) is a knowledge-base of *frames*, describing prototypical situations. Frames can be related to each other by inter-frame relations, e.g. *Inheritance*, *Precedence*, *Usage* and *Perspective*.

For each frame, several semantic roles are specified, called *frame elements* (FEs), denoting the participants in the situation described. Each FE may be labeled as *core* if it is central to the frame. For example, some core FEs of the *Commerce_pay* frame are *Buyer* and *Goods*, while a non-core FE is *Place*. Each FE may also be labeled with a semantic type, e.g. *Sentient*, *Event*, and *Time*.

A frame includes a list of predicates that can evoke the described situation, called *lexical units* (LUs). LUs are mainly verbs but may also be nouns or adjectives. For example, the frame *Commerce_pay* lists the LUs *pay.v* and *payment.n*.

Finally, FrameNet contains annotated sentences that represent typical LU occurrences in texts. Each annotation refers to one LU in a specific frame and the FEs of the frame that occur in the sentence. An example sentence is *"I_{Buyer} have to pay the $bills_{Money}$"*. Each sentence is accompanied by a *valence pattern*, which provides, among other info, grammatical functions of the core FEs with respect to the LU. The valence pattern of the above sentence is *[(Buyer Subj), (Money Obj)]*.

2.3 Using FrameNet for Semantic Inference

To the best of our knowledge, the only work that utilized FrameNet for entailment rule generation is LexPar (Coyne and Rambow, 2009). LexPar first identifies lexical entailment relations by going over all LU pairs which are either in the same frame or whose frames are related by one of FrameNet's inter-frame relations. Each candidate pair is considered entailing if the two LUs are either synonyms or in a direct hypernymy relation in WordNet (providing the vast majority of LexPar's relations), or if their related frames are connected via the *Perspective* relation in FrameNet.

Then, argument mappings between each entailing LU pair are extracted based on the core FEs that are shared between the two LUs. The syntactic positions of the shared FEs are taken from the valence patterns of the LUs. A LexPar rule example is presented in Figure 3 (top part).

Since most of LexPar's entailment relations are based on WordNet's relations, LexPar's rules could be viewed as an intersection of WordNet and FrameNet lexical relations, accompanied with argument mappings taken from FrameNet.

3 Rule Extraction from FrameNet

The above prior work identified lexical entailment relations mainly from WordNet, which limits the use of FrameNet in two ways. First, some relations that appear in FrameNet are missed because they do not appear in WordNet. Second, unlike FrameNet, WordNet does not include argument mappings for its relations. Thus, prior work for rule generation considered only substitutable relations from WordNet (*synonyms* and *hypernyms*), not utilizing FrameNet's capability to map arguments of non-substitutable relations.

Our goal in this paper is to generate entailment rules solely from the information within FrameNet. We present a novel algorithm for generating entailment rules from FrameNet, called *FRED* (**Fr**ameNet **E**ntailment-rule **D**erivation), which operates in three steps: a) extracting templates for each LU; b) detecting lexical entailment relations between pairs of LUs; c) generating entailment rules by mapping the arguments between two LUs in each entailing pair.

3.1 Template Extraction

Many LUs in FrameNet are accompanied by annotated sentences (Section 2.2). From each sentence of a given LU, we extract one template for each annotated FE in the sentence. Each template includes the LU, one argument corresponding to the target FE and their syntactic relation in the sentence parse-tree. We focus on extracting unary templates, as they can describe any ar-

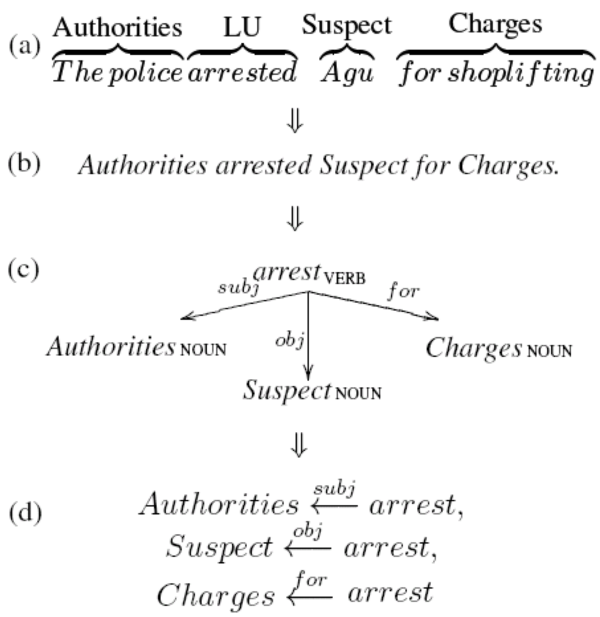

Figure 1: Template extraction for a sentence containing the LU *'arrest'*.

gument mapping by decomposing templates with several arguments into unary ones (Szpektor and Dagan, 2008). Figure 1 exemplifies this process.

As a pre-parsing step, all FE phrases in a given sentence are replaced by their related FE names, excluding syntactic information such as prepositions or possessives (step (b) in Figure 1). Then, the sentence is parsed using the Minipar dependency parser (Lin, 1998) (step (c)). Finally, a path in the parse-tree is extracted between each FE node and the node of the LU (step (d)). Each extracted path is converted into a template by replacing the FE node with an argument variable.

We simplify each extracted path by removing nodes along the path that are not part of the syntactic relation between the LU and the FE, such as conjunctions and other FE nodes. For example, '$Authorities \stackrel{subj}{\longleftarrow}$ enter $\stackrel{conj}{\longrightarrow}$ *arrest'* is simplified into '$Authorities \stackrel{subj}{\longleftarrow}$ *arrest'*.

Some templates originated from different annotated sentences share the same LU and syntactic structure, but differ in their FEs. Usually, one of these templates is incorrect, due to erroneous parse (e.g. '$Suspect \stackrel{obj}{\longleftarrow}$ *arrest'* is a correct template, in contrast to '$Charges \stackrel{obj}{\longleftarrow}$ *arrest'*). We thus keep only the most frequently annotated template out of the identical templates, assuming it is the correct one.

3.2 Identifying Lexical Entailment Relations

FrameNet groups LUs in frames and describes relations between frames. However, relations between LUs are not explicitly defined. We next describe how we automatically extract several types of lexical entailment relations between LUs using two approaches.

In the first approach, LUs in the same frame that are morphological derivations of each other, e.g. *'negotiation.n'* and *'negotiate.v'*, are marked as paraphrases. We take morphological derivation information from the CATVAR database (Habash and Dorr, 2003).

The second approach is based on our observation that some LUs express the prototypical situation that their frame describes, which we denote *dominant LUs*. For example, the LU *'recover'* is dominant for the *Recovery* frame. We mark LUs as dominant if they are morphologically derived from the frame's name.

Our assumption is that since dominant LUs express the frame's generic meaning, their meaning is likely to be entailed by the other LUs in this frame. Consequently, we generate such lexical rules between any dominant LU and any other LU in a given frame, e.g. '*heal* ⇒ *recover*' and '*convalescence* ⇒ *recover*' for the *Recovery* frame.

In addition, we assume that if two frames are related by some type of entailment relation, their dominant LUs are also related by the same relation. Accordingly, we extract entailment relations between dominant LUs of frames that are connected via the *Inheritance*, *Cause* and *Perspective* relations, where *Inheritance* and *Cause* generate directional entailment relations (e.g. '*choose* ⇒ *decide*' and '*cure* ⇒ *recover*', respectively) while *Perspective* generates bidirectional paraphrase relations (e.g. '*transfer* ⇔ *receive*').

Finally, we generate the transitive closure of the set of lexical relations identified by the above methods. For example, the combination of '*sell* ⇔ *buy*' and '*buy* ⇒ *get*' generates '*sell* ⇒ *get*'.

3.3 Generating Entailment Rules

The final step in the FRED algorithm generates lexical syntactic entailment rules from the extracted templates and lexical entailment relations.

For each identified lexical relation '*left* ⇒ *right*' between two LUs, the set of FEs that are shared by both LUs is collected. Then, for each shared FE, we take the list of templates that connect this FE

Lexical Relation:
cure $\Rightarrow$ recovery

Templates:

$Patient \overset{obj}{\longleftarrow}$ *cure*	(cure *Patient*)
$Affliction \overset{of}{\longleftarrow}$ *cure*	(cure of *Affliction*)
$Patient \overset{gen}{\longleftarrow}$ *recovery*	(*Patient*'s recovery)
$Patient \overset{of}{\longleftarrow}$ *recovery*	(recovery of *Patient*)
$Affliction \overset{from}{\longleftarrow}$ *recovery*	(recovery from *Affliction*)

Intra-LU Entailment Rules:
$Patient \overset{gen}{\longleftarrow}$ *recovery* $\Longleftrightarrow$ $Patient \overset{of}{\longleftarrow}$ *recovery*

Inter-LU Entailment Rules:
$Patient \overset{obj}{\longleftarrow}$ *cure* $\Longrightarrow$ $Patient \overset{gen}{\longleftarrow}$ *recovery*
$Patient \overset{obj}{\longleftarrow}$ *cure* $\Longrightarrow$ $Patient \overset{of}{\longleftarrow}$ *recovery*
$Affliction \overset{of}{\longleftarrow}$ *cure* $\Longrightarrow$ $Affliction \overset{from}{\longleftarrow}$ *recovery*

Figure 2: Some entailment rules generated for the lexical relation '*cure.v* $\Rightarrow$ *recovery.n*'.

Configuration	R (%)	P (%)	F1
No-Rules	13.8	57.7	20.9
LexPar	14.1	42.9	17.4
WordNet	18.3	32.2	17.8
FRED	17.6	55.1	24.6
FRED $\cup$ WordNet	21.8	33.3	20.9

Table 1: Macro average Recall (R), Precision (P) and F1 results for the tested configurations.

to each of the LUs, denoted by T^{fe}_{left} and T^{fe}_{right}. Finally, for each template pair, $l \in T^{fe}_{left}$ and $r \in T^{fe}_{right}$, the rule '$l \Rightarrow r$' is generated. In addition, we generate paraphrase rules between the various templates including the same FE and the same LU. Figure 2 illustrates this process.

To improve rule quality, we filter out rules that map FEs of adjunct-like semantic types, such as *Time* and *Location*, since different templates of such FEs may have different semantic meanings (e.g. '$Time \overset{before}{\longleftarrow}$ *arrive*' '$Time \overset{after}{\longleftarrow}$ *arrive*'). Thus, it is hard to identify those template pairs that correctly map these FEs for entailment.

We manually evaluated a random sample of 250 rules from the resulting rule-set, out of which we judged 69% as correct.

4 Application-based Evaluation

4.1 Experimental Setup

We would like to evaluate the overall utility of our resource for NLP applications, assessing the correctness of the actual rule applications performed in practice, as well as to compare its performance to related resources. To this end, we follow the experimental setup presented in (Szpektor and Dagan, 2009), which utilized the ACE 2005 event dataset[3] as a testbed for entailment rule-sets. We briefly describe this setup here.

The task is to extract argument mentions for 26 events, such as *Sue* and *Attack*, from the ACE annotated corpus, using a given tested entailment rule-set. Each event is represented by a set of unary *seed templates*, one for each event argument. Some seed templates for *Attack* are '*Attacker*$\overset{subj}{\longleftarrow}$attack' and 'attack$\overset{obj}{\longrightarrow}$*Target*'.

Argument mentions are found in the ACE corpus by matching either the seed templates or templates entailing them found in the tested rule-set. We manually added for each event its relevant WordNet synset-ids and FrameNet frame-ids, so only rules fitting the event target meaning will be extracted from the tested rule-sets.

4.2 Tested Configurations

We evaluated several rule-set configurations:

No-Rules The system matches only the seed templates directly, without any additional rules.

WordNet Rules are generated from WordNet 3.0, using only the *synonymy* and *hypernymy* relations (see Section 2.1). Transitive chaining of relations is allowed (Moldovan and Novischi, 2002).

LexPar Rules are generated from the publicly available LexPar database. We generated unary rules from each LexPar rule based on a manually constructed mapping from FrameNet grammatical functions to Minipar dependency relations. Figure 3 presents an example of this procedure.

FRED Rules are generated by our algorithm.

FRED $\cup$ WordNet The union of the rule-sets of FRED and WordNet.

4.3 Results

Each configuration was tested on each ACE event. We measured *recall*, *precision* and *F1*. Table 1 reports macro averages of the three measures over the 26 ACE events.

As expected, using *No-Rules* achieves the highest precision and the lowest recall compared to all other configurations. When adding LexPar rules,

[3] `http://projects.ldc.upenn.edu/ace/`

LexPar rule:
Lexemes: arrest $\longrightarrow$ apprehend
Valencies: [(Authorities Subj), (Suspect Obj), (Offense (for))] $\Longrightarrow$ [(Authorities Subj), (Suspect Obj), (Offense (in))]

Generated unary rules:
$X \overset{subj}{\longleftarrow}$ arrest $\Longrightarrow X \overset{subj}{\longleftarrow}$ apprehend , arrest $\overset{obj}{\longrightarrow} Y \Longrightarrow$ apprehend $\overset{obj}{\longrightarrow} Y$, arrest $\overset{for}{\longrightarrow} Z \Longrightarrow$ apprehend $\overset{in}{\longrightarrow} Z$

Figure 3: An example for generation of unary entailment rules from a LexPar rule.

only a slight increase in recall is gained. This shows that the subset of WordNet rules captured by LexPar (Section 2.3) might be too small for the ACE application setting.

When using all WordNet's substitutable relations, a substantial relative increase in recall is achieved (32%). Yet, precision decreases dramatically (relative decrease of 44%), causing an overall decrease in F1. Most errors are due to correct WordNet rules whose LHS is ambiguous. Since we do not apply a WSD module, these rules are also incorrectly applied to other senses of the LHS. While this phenomenon is common to all rule-sets, WordNet suffers from it the most since it contains many infrequent word senses.

Our main result is that using FRED's rule-set, recall increases significantly, a relative increase of 27% compared to No-Rules, while precision hardly decreases. Hence, overall F1 is the highest compared to all other configurations (a relative increase of 17% compared to No-Rules). The improvement in F1 is statistically significant compared to all other configurations, according to the two-sided Wilcoxon signed rank test at the level of 0.01 (Wilcoxon, 1945).

FRED preforms significantly better than LexPar in both recall, precision and F1 (a relative increase of 25%, 28% and 41% respectively). For example, LexPar hardly utilizes FrameNet's argument mapping capabilities since most of its rules are based on a sub-set of WordNet's substitutable relations.

FRED's precision is substantially higher than WordNet. This mostly results from the fact that FrameNet mainly contains common senses of predicates while WordNet includes many rare word senses; which, as said above, harms precision when WSD is not applied. Error analysis showed that only 7.5% of incorrect extractions are due to erronous rules in FRED, while the majority of errors are due to sense mismatch or syntactic matching errors of the seed templates ot entailing templates in texts.

FRED's Recall is somewhat lower than WordNet, since FrameNet is a much smaller resource. Yet, its rules are mostly complementary to those from WordNet. This added value is demonstrated by the 19% recall increase for the union of FRED and WordNet rule-sets compared to WordNet alone. FRED provides mainly argument mappings for non-substitutable WordNet relations, e.g. '*attack.n on X* $\Rightarrow$ *attack.v X*', but also lexical relations that are missing from WordNet, e.g. '*ambush.v* $\Rightarrow$ *attack.v*'.

Overall, our experiment shows that the rule-base generated by FRED seems an appropriate complementary resource to the widely used WordNet-based rules in semantic inference and expansion over predicates. This suggestion is especially appealing since our rule-set performs well even when a WSD module is not applied.

5 Conclusions

We presented FRED, a novel algorithm for generating entailment rules solely from the information contained in FrameNet. Our experiment showed that FRED's rules perform substantially better than LexPar, the only prior rule-set derived from FrameNet. In addition, FRED's rule-set largely complements the rules generated from WordNet because it contains argument mappings between non-substitutable predicates, which are missing from WordNet, as well as lexical relations that are not included in WordNet.

In future work we plan to investigate combining FrameNet and WordNet rule-sets in a transitive manner, instead of their simple union.

Acknowledgments

This work was partially supported by the Rector's research grant of Bar-Ilan University, the PASCAL-2 Network of Excellence of the European Community FP7-ICT-2007-1-216886 and the Israel Science Foundation grant 1112/08.

References

Collin Baker, Charles Fillmore, and John Lowe. 1998. The berkeley framenet project. In *Proceedings of COLING-ACL*, Montreal, Canada.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Second PASCAL Challenge Workshop for Recognizing Textual Entailment*.

Roni Ben Aharon. 2010. Generating entailment rules from framenet. Master's thesis, Bar-Ilan University.

Robert Coyne and Owen Rambow. 2009. Lexpar: A freely available english paraphrase lexicon automatically extracted from framenet. In *Proceedings of the Third IEEE International Conference on Semantic Computing*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Lecture Notes in Computer Science*, volume 3944, pages 177–190.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for english. In *Proceedings of the North American Association for Computational Linguistics (NAACL '03)*, pages 96–102, Edmonton, Canada. Association for Computational Linguistics.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC*.

Dan Moldovan and Adrian Novischi. 2002. Lexical chains for question answering. In *Proceedings of COLING*.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK, August.

Idan Szpektor and Ido Dagan. 2009. Augmenting wordnet-based inference with argument mapping. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 27–35, Suntec, Singapore, August.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.