

Generating FrameNets of various granularities: The FrameNet Transformer

Josef Ruppenhofer, Jonas Sunde, Manfred Pinkal

Saarland University
Saarbrücken, Germany
{josefr, jsunde, pinkal}@coli.uni-saarland.de

Abstract

We present a method and a software tool, the FrameNet Transformer, for deriving customized versions of the FrameNet database based on frame and frame element relations. The FrameNet Transformer allows users to iteratively coarsen the FrameNet sense inventory in two ways. First, the tool can merge entire frames that are related by user-specified relations. Second, it can merge word senses that belong to frames related by specified relations. Both methods can be interleaved. The Transformer automatically outputs format-compliant FrameNet versions, including modified corpus annotation files that can be used for automatic processing. The customized FrameNet versions can be used to determine which granularity is suitable for particular applications. In our evaluation of the tool, we show that our method increases accuracy of statistical semantic parsers by reducing the number of word-senses (frames) per lemma, and increasing the number of annotated sentences per lexical unit and frame. We further show in an experiment on the FATE corpus that by coarsening FrameNet we do not incur a significant loss of information that is relevant to the Recognizing Textual Entailment task.

1. Introduction

The level of predicate-argument structure has proven essential for various NLP applications. The most prominent resources for modelling predicate-argument structure in English are PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998). While PropBank focusses on the mapping of different syntactic realizations of one lemma to the same predicate-argument structure, using lemma-specific semantic roles, FrameNet’s word senses, called lexical units (LU), are grouped into frames. FrameNet’s semantic roles, called Frame Elements, are defined at the level of frames which relate word senses of different lemmas to each other. Moreover, both frames and frame elements are also organized in a hierarchy using several types of semantic relations. The detail and richness of FrameNet seems desirable for information access tasks such as recognizing textual entailment, information extraction, and question answering. However, it is often difficult to use the FrameNet data to advantage, as discussed e.g. by Shen & Lapata 2007 for question answering. The major obstacle is the modest performance of shallow semantic parsers. Burchardt et al. 2009 report results on the recognizing textual entailment (RTE) task showing that FrameNet information has the potential to significantly increase the discriminative power of RTE systems, as compared to PropBank information or simple word overlap measures. However, this increase is almost completely offset by the inaccuracy of the parser.

In this paper, we will address two interdependent reasons for the lack of performance of FrameNet based frame and role labellers. First, many frames, frame elements and lexical units are exemplified by relatively few annotated training instances in the FrameNet data (e.g. Kaiser & Webber 2007). In fact, some word senses are not illustrated by any annotations. Second, distinctions between frames and frame elements are often too fine-grained (Burchardt et al. 2009) to allow robust shallow semantic parsing. FrameNet distinguishes, for instance, between a *Location_of_light* frame and a *Light_movement*

frame. As a result, many lemmas like *glow* belong both to the former (1) and the latter (2) frame, making them polysemous.

- (1) [The walls ^{*Figure*} **glowed** [in the sun ^{*Ground*}].
- (2) [The evening sun ^{*Emitter*} **glowed** [from the west ^{*Source*} [as brassy as a dinner-gong ^{*Depictive*}].

According to the definitions, while the *Location_of_light* frame focuses on a perceptible Figure showing up against a *Ground* location due to *Light* shining on/from the *Ground*, the *Light_movement* frame is concerned with a scene where an *Emitter* emits a *Beam* of light from a *Source*, along a *Path*, and/or towards a *Goal*. The distinction between the two frames is difficult both for semantic parsers and human annotators.

We present a parameterizable tool, the FrameNet transformer, for (semi-)automatically deriving customized versions of the FrameNet database in which frames and/or lexical units are merged based on frame and frame element relations. The resulting alternative FrameNets have fewer different frames and more annotated instances for the merged frames and frame elements, and we can use them to determine experimentally which level of granularity makes shallow semantic parsing most robust while still preserving the discriminative semantic power needed, for instance, in the context of the entailment recognition task.

The FN transformer has two major modes for reconfiguring the FrameNet data. (A) In *relations* mode, it merges frames that are related to each other via a user-specifiable set of frame relations. (B) In *lexical unit* mode, it only merges word senses that are related in the FrameNet hierarchy without altering the structure of the frame hierarchy. The FN transformer is implemented in Java, relies only on publicly available components, and will be made freely available to the research community.

Our paper is structured as follows. In section 2., we discuss related work. Section 3. presents our data and method. Section 4. describes the functionality and implementation

of the FN Transformer. In section 5, we present an evaluation of our tool before offering conclusions in section 6..

2. Related Work

Some researchers have worked on parsing performance by addressing the coverage problem which results from the incompleteness of the FrameNet database. Burchardt et al. (2005) present a rule-based system that uses WordNet to generalize over a given target word in order to assign frames to words unknown to FrameNet. Fürstenau and Lapata (2009a) address the same problem with a semi-supervised learning approach which acquires training instances for unseen verbs from an unlabeled corpus.

Fürstenau and Lapata (2009b) use a similar data expansion strategy to augment the training material for known verbs. Matsubayashi et al. (2009) focus on the role assignment problem. They do not produce additional training instances, but rather compare various ways of generalizing across semantic roles, assuming that the frame evoking word, the frame, and the set of argument phrases to be labeled are given. Some of Matsubayashi et al.’s methods are potentially noisy as they generalize roles across frames that are not connected (closely) in the hierarchy by relying on the similarity of frame element names, identity of semantic type assigned to roles, or common mappings to the same VerbNet thematic role.

The approach of McConville and Dzikovska (2008) is most similar to ours. They produce a complete semantic resource by deriving a verb lexicon for deep syntactic parsing from FrameNet’s annotations using the *Inheritance* relation between frames and the *CoreSet* relation between Frame Elements to reduce the set of semantic roles. Their lexicon comes, however, without a corresponding set of annotated sentences, and is thus not usable for training shallow semantic parsers, while the output of our transformation tool includes an appropriately updated corpus.

3. Method

The FN Transformer offers two modes for transforming FrameNet data, a *relation mode* and a *lexical unit mode*. In the former, whole frames are merged into others if they are connected by specific types of frame relations. In the latter, the frame inventory remains unchanged, but some word senses are merged into others. Other modes of merging are conceivable. For instance, one could try to reassign individual annotations from one word sense to another in the hope of creating sharper distinctions, or one could merge frames whose Frame Elements have similar names, or frames whose lexical units share lemmas. The two modes the transformer uses have some advantages over these alternatives, however. They exploit explicit human judgments that were encoded in the frame definitions, frame relations, and sense distinctions by experts. They do not rely on additional, potentially noisy, mechanisms to determine frames or annotation instances that are similar. When merging frames in relation-mode, we are also able to redirect, in a meaningful way, frame relations from frames that disappear to the ones that absorb their lexical units.

3.1. Relation mode

The relation mode considers the types of relations that connect two frames in the FrameNet hierarchy in deciding whether a pair of frames should be merged. Users of the tool control which relation types are taken into account. The tool uses relations to identify *target frames*, that is the set of frames into which other frames can be merged. Specifically, target frames are those that are parents in instances of the specified relation types. The tool also uses user-specified relations to identify *source frames* that should be merged into target frames. A source frame is one which can be reached from a target frame via one of the specified relations and below which there are no other frames that are reachable by one of the specified relations. Typically, the same set of relations is used both for identifying target frames and for finding source frames but users can specify distinct sets of relations for the two purposes.

In a single run of the transformer, the lexical units of a source frame being merged away move up exactly one step into their direct parent. Lexical units in sibling source frames move into their parent frame together. The merging process can be run iteratively, so that additional frames, which have newly become source frames, can be merged upwards into other target frames.

After each iteration, the tool produces a new “release”, that is, it propagates the changes in the frame and frame element hierarchy to all relevant XML-files of the FrameNet database release. This involves automatically changing labels and moving annotations based on the mappings between frame elements that FrameNet specifies as part of its frame-to-frame relations. The frame relations of eliminated source frames are redirected, too.

In the process of relabelling and moving annotations, the following problems must be considered:

- In certain cases, the annotations to be moved include labels that are not defined for the frame they move to. When this happens, the tool will newly introduce the labels into the target frame, but only if the label is non-core. For instance, the *Membership* frame includes a non-core frame element *Standing*, which the *Becoming_a_member* frame lacks. To preserve the annotations of words like *member* and *membership* in the move from *Membership* to *Becoming_a_member* via the *INCHOATIVE_OF* relation, the tool will introduce the FE *Standing* into the *Becoming_a_member* frame. In cases where the target frame lacks Core Frame Elements of the source frame, the merger still takes place but the extra core FEs of the source frame will not be added to the target frame. The user is alerted to such cases in the log file.
- The source frame may include a Frame Element that has no explicit mapping to the target frame in the FrameNet hierarchy but the target frame includes an FE of the same name. In such cases, the tool will map the two FEs onto each other.

Our current work is based on FrameNet release 1.3, with 795 frames and 10195 lexical units. A trial run that used *SUBFRAME*, *USING*, *CAUSATIVE_OF*, and

INCHOATIVE_OF to identify target frames and used these four relation types plus INHERITANCE as traversable relations eliminated 297 frames, which is more than a third of the total. Another run with the same setup except that INHERITANCE was excluded from the set of traversable relations eliminated 134 frames.

As an example of a relation-based adjustment, consider Figures 1, which shows a subpart of the frames having to do with crime. If we use the SUBFRAME, PERSPECTIVE_ON, CAUSATIVE_OF and INCHOATIVE_OF relations to identify starting nodes and as the traversable relations, we will see the frames and frame relations change as shown by Figures 2 and 3 over the course of two iterations. The *Committing_crime* and *Criminal_investigation* frames disappear on the first iteration, their lexical units having moved to *Crime_scenario*. The 8 child frames of *Committing_crime* in INHERITANCE and USING relationships are reattached to *Crime_scenario*. The USING relationship between *Criminal_investigation* and *Suspicion* is redirected to *Crime_scenario*. Also on the first iteration, the sub-frames of *Arraignment* and *Trial* are folded into their respective parent frames. Of the child frames of *Committing_crime*, only *Arrest* is folded into the parent. Its USING relationship to *Surrendering* is redirected to *Committing_crime*. The *Arraignment* and *Trial* frames are absorbed by their parent *Committing_crime* only on the second iteration, when they no longer have frames beneath them that are reachable by traversable relations.

3.2. Lexical unit mode

In lexical unit mode, the FN transformer merges and migrates related (frame-specific) senses of a particular lemma, rather than frames as a whole. Lexical units can be related in two basically different ways: In the more special case, one LU is the ancestor of the other, more specific LU in the FrameNet hierarchy. Merging the two would amount to adjusting the annotations of the latter, the source LU, to those of the more general LU, the target LU. The more general case is that neither LU dominates the other. In this case, neither of the original LUs would persist. Instead we would create a new LU in a third frame, reflecting the broader semantic range covered by the combination.

In the current implementation of the tool, only the simple case is dealt with. Specifically, if the potential target LU is in a frame that is the closest ancestor reachable from the frame of the potential source LU, then we adjust the annotations of the source LU and reassign them to the target LU. Subsequently, we eliminate the source LU. As in relation mode, the user must specify the acceptable kinds of relations along the path from potential source to potential target LUs. However, different from relation mode, relation types are not used in identifying target LUs. The reason is that lexical unit modes requires subsumption and lemma-identity between lexical units that are candidates for merging. There is thus no danger that over a number of iterations a single target LU might absorb excessively many weakly related source LUs.

Lemma	Target frame	Source frame
cook.v	Apply heat	Cooking creation
chuckle.v	Make noise	Comm. noise
bray.v	Make noise	Comm. noise
transformation.n	Cause change	Undergo change
stuff.v	Placing	Filling
cram.v	Placing	Filling
bolt.v	Departing	Quitting a place
search.v	Scrutiny	Seeking
hair.n	Observable bodyparts	Hair configuration
ecoterrorism.n	Terrorism	Terrorism

Table 1: Lemmas where one lexical unit is the ancestor of another

A typical example of an LU-based merger is that between the LU *cook.v* in the *Apply_heat* frame and the LU *cook.v* in the *Absorb_heat* frame, assuming that the CAUSATIVE_OF relation between them is defined as traversable. In some rare cases, it is possible that a source LU has more than one possible target LU that it could be moved to. In such cases, the tool by default prefers merging across shorter distances and disprefers merging across a USING relation. If these preferences do not lead to a unique merger to perform, the tool chooses randomly from among the possible best mergers. Users can change the default behavior in the settings file.

FrameNet release 1.3 has 1316 lemmas that occur in more than one frame. Mostly, they have 2 known senses but some belong to as many as 9 different frames. In total, there are 2587 pairs of senses that could potentially be merged. For 530 of these pairs, one of the two lexical units is subsumed by the other. Some examples are given in Table 1.

3.3. Choosing relations for relation-based merging

In relation mode, the choice of frame relation types determines the selection of target and source frames for merging. Frames related by PERSPECTIVE_ON, SUBFRAME (plus PRECEDES), CAUSATIVE_OF and INCHOATIVE_OF relations are good candidates for merging as those relations indicate reliably close semantic relations. The PERSPECTIVE_ON relation is used to relate different points-of-view of a situation to a Neutral frame that expresses a sort of god’s eye view on the situation. Thus, a Neutral frame generally has at least two perspectivized frames. An example where the PERSPECTIVE_ON relation comes into play is employment, where FrameNet has separate frames for the perspectives of an employer and an employee.

The SUBFRAME relation is used to connect sequences of states and transitions, each of which can itself be separately described as a frame. The separate frames (called subframes) are related to the complex frames via the SUBFRAME relation. Frame elements of the complex frame may be identified (mapped) to the frame elements of the subparts. But the sets of frame elements are typically not fully identical between the parent frame and any of its children, nor between the children. For instance, the figure of the criminal occurs, under various names, in all of the frames connected to the *Criminal_process*. By contrast, the notion of a judicial *Sentence* occurs only

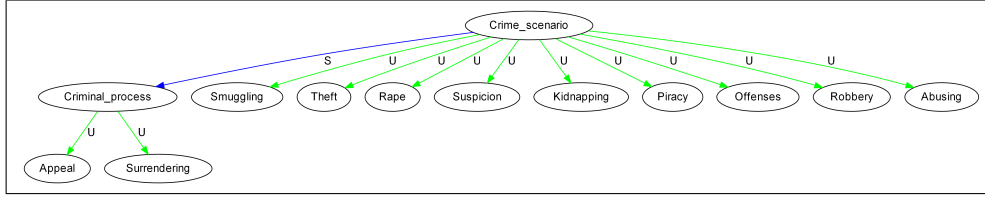


Figure 3: Crime scenario after 2 iterations

But consider now the pair *Transitive_action* and *Cause_to_end*, which are also related as parent and child via *INHERITANCE*. *Cause_to_end* has only one core FE that has no counterpart in *Transitive_action* and the two frames are similar in their non-core FEs. But intuitively it is clear that the distance between the two frames is great. In fact, while *Cause_to_end* has lexical units associated with it (*arrest.v*, *end.v*, *lift.v*, *lifting.n*, *put an end to.v*), *Transitive_action* is an abstract, so-called non-lexical frame.

The *USING* relation presents a similar difficulty. It is intentionally defined to be a rather vague relation type to be used in cases where understanding one frame requires some sort of appreciation of another. As an example of great similarity, consider the *Emptying* frame and the *Removing* frame that it “Uses”. The core FEs map perfectly and the non-core FEs are also very similar. In addition, the *Emptying* frame shares a quarter of its lexical units (10) with the *Removing* frame. As a contrast to the *Emptying-Removing* pair, consider the *Roadways* frame and the *Motion* frame that it “Uses”. These share no LUs and they are semantically very different, with the former containing only nouns denoting physical paths and the latter only verbs that can express some very schematic sort of motion. Thus, we would not want to merge these two frames.

As the examples show, the automatic choice criteria provided by relation types are too coarse-grained. To allow for the merger of frame pairs connected by *INHERITANCE* or *USING*, our tool supports also the manual specification of frame pairs. Another user-controlled mechanism that the tool provides is to specify “stop frames”, that is, frames that other frames should not be merged into. Using stop frames makes sense for abstract frames high up in the hierarchy, e.g. the *Event* frame, since otherwise semantically very different frames might get merged into them.

4. The FrameNet Transformer

The *FrameNetTransformer* is written in Java 1.6. As described earlier, it supports two major modes for reconfiguring *FrameNet*, relation-based and lexical unit-based. It currently has no GUI but simply loads user settings from xml files. The basic settings for both modes include path specifications to the *FrameNet* data release, to an output directory, and to a logfile to be created.

In relation-based mode, frame relations are used to compute candidate frames for merging. The user can set the relations that may be traversed to find starting points and merging candidates as well as frames which cannot be passed on the way (*stop frames*). The output of relation-based merging is a changed *FrameNet* Version, .dot files that can be used to render images (via the open-source *GraphViz* software)

to illustrate the changes effected by the transformer, and a log file with all relevant information. Figure 4 is an example of the tool’s graphical output. The red lines show that as a result of a run of the tool, the *INCHOATIVE_OF* relation between *Becoming a member* and *Membership* was eliminated along with the *Using* relation between *Membership* and *Exclude_member*. The green line indicates that as a result of the run, a new *USING* link was created between the *Becoming a member* frame and the *Exclude_member* frame.

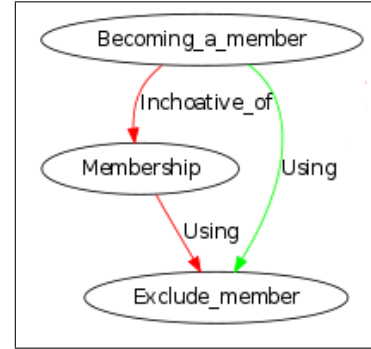


Figure 4: Membership frames

In lexical-unit mode, lexical units that belong to the same lemma constitute candidates for merging. The user can specify a cost for the traversal of each frame relation type so as to resolve conflicts in cases where a word sense may be merged into more than one other word sense. The output of this mode is a changed *FrameNet* Version and a log file with all relevant information.

As the *FN Transformer* is mainly intended for use in the context of automatic processing, the tool omits certain file types from its output that are meant for human users. In particular, it does not output HTML annotation files, nor does it update lexical entry report files. Similarly, while the tool can reassign word senses to other frames, it cannot automatically adjust frame definitions so as to make them match the changed set of associated lexical units.

5. Evaluation

The *FN transformer* efficiently generates coarser-grained variants of the *FrameNet* database. The transformation reduces the number of word-senses (frames) per lemma, and increases the number of annotated sentences per lexical unit and frame. A baseline evaluation thus consists in confirming that we do indeed obtain improved accuracy of frame-semantic parsers trained on the modified data.

In a further step, we check whether we improve parsing accuracy at the cost of losing relevant information.

What relevant information is can only be defined in a task-specific context. In this paper, we study the contribution of FrameNet information to the RTE-task, a meta-task for information access applications such as Question Answering, Information Extraction, Information Retrieval, and Summarization.

5.1. Impact on parsing accuracy

It may appear straightforward that coarsening of FN frames would improve parsing accuracy. However, we cannot take this for granted. First, if the merging of frames and/or lexical units merges two word senses that are both very infrequent in our training data, then we are likely not to see a benefit in the overall results because statistical parsers such as the Shalmaneser SRL system (Erk & Pado (2006)) that we use typically suffer from the class imbalance problem. That is, if the overall distribution is very skewed, the system does not learn low-frequency classes.

Finally, our expectation that we should see a benefit by merging along linguistically well motivated types of relations rests on the assumption that the sense distinctions that are hard for human annotators are also the ones that are difficult for semantic parsers. But this is not necessarily so. The results of an annotation experiment by Rehbein et al. ((2009)), for instance, show that the Shalmaneser system sometimes makes errors on word senses that humans never confuse. For instance, humans never mixed up the *Perception_active* and *Appearance* senses of *look*, shown in (5-6), whereas Shalmaneser did.

(5) He **looked** across at Frau Nordern .

(6) Kim’s new furniture **looks** sort of bluish.

5.1.1. Experimental setup

To evaluate the impact of frame or lexical unit merging on the performance of an automatic semantic role labeler, we compared the performance of the Shalmaneser system in two settings. As a baseline, we trained and tested the system on original FrameNet data. We then trained and tested Shalmaneser on data output by our transformer. We did not compare performance on the full FrameNet but only on the subset of lemmas that were affected by the transformation, that is, lemmas where annotations of at least one of their lexical units (=word senses) were reassigned to another frame. All experiments were carried out in a 10-fold cross-validation setting on data from FrameNet release 1.3. On each kind of dataset, we evaluated frame assignment, argument recognition, and argument labeling. Frame assignment is the task of word sense disambiguation. Argument recognition is the classification of constituents as arguments of the frame evoking word. Argument labeling is the task of assigning the correct label to a constituent correctly recognized as an argument.

For relation-based merging we used a transformed FN, henceforth called FN1.3R, produced with the following settings. We used the CAUSATIVE_OF, INCHOATIVE_OF, PERSPECTIVE_ON and SUBFRAME relations both to define starting frames and as traversable relations. We defined the following frames as stop frames: *Event*, *Process*, *State*, *Activity*, *Transitive_action*,

Reciprocity, *Trajector-Landmark*, *Intentionally_affect*, *Intentionally_act*, *Artifact*, *Entity*, *Physical_entity*. We performed 2 iterations of merging because we knew from an earlier experiment with the same relations chosen that all possible frame reassignments take place within the first 3 iterations, and we wanted to be conservative. 1010 distinct lemmas with 1460 associated lexical units were affected by frame (and word sense) mergers.

For lexical unit-based merging we use a transformed FN, henceforth called FN1.3LU, produced with the following settings. We defined CAUSATIVE_OF, INCHOATIVE_OF, PERSPECTIVE_ON, SUBFRAME as traversable relations. We performed a single iteration, as a result of which 485 pairs of word senses were merged. These word sense pairs came from 450 distinct lemmas.

5.1.2. Results

The results are shown in Table 2 for relation-based merging and in Table 3 for lexical unit-based merging. The tables show the mean accuracy across the ten folds for each task separately under the heading *task*, and the cumulative performance for all prior tasks and the current task under the heading *cum.*

	FN1.3		FN1.3R	
	task	cum.	task	cum.
Frame assignment	0.94	0.94	0.94	0.94
Argument recognition	0.69	0.64	0.69	0.65
Argument labeling	0.71	0.46	0.75	0.49

Table 2: Performance of Shalmaneser on FN release 1.3 and relation-based transformation (10-fold cross-validation)

	FN1.3		FN1.3LU	
	task	cum.	task	cum.
Frame assignment	0.89	0.89	0.94	0.94
Argument recognition	0.69	0.62	0.66	0.62
Argument labeling	0.74	0.46	0.72	0.44

Table 3: Performance of Shalmaneser on FN release 1.3 and lu-based transformation (10-fold cross-validation)

5.1.3. Discussion

The results in Table 2 show that for relation-based merging, the collapsing of certain frame distinctions leaves the performance on frame assignment unaffected. Likewise, the performance on the argument recognition task remains the same as on the original data. The performance on argument labeling is improved, which may be a benefit of the fact that merging increases the number of instances per label that are available for training. The cumulative performance of the system on the transformed data is better than on the original data, resulting in more correctly labeled role instances.

Table 3 shows that, for lexical unit-based merging, the performance on frame assignment improves but that on argument recognition drops slightly. Unlike for relation-based

merging, argument labeling suffers a slight loss in performance on the transformed data. The reason for this is most likely that the lexical-unit based merging collapses many causative-inchoative pairs of verbs. These verbs differ systematically in how they map certain core semantic roles to the syntax, as shown by (7-8). Merging their instances, makes it harder to associate grammatical functions (or paths in trees) with semantic roles.

(7) He **dried** [his coat *Dryee*] by the fire.

(8) [The coat *Dryee*] **dried** by the fire.

On the data transformed by lexical unit-based merging, the cumulative performance of the Shalmaneser parser is slightly worse than on the original FN data. But on the data transformed by relation-based merging, parser performance is slightly better than on the original data.

5.2. Preservation of relevant information

We want to ensure that better parser performance is not achieved at the cost of losing relevant information needed for specific applications. To this end, we evaluate our coarsened FrameNet versions in the context of the entailment recognition task. Entailment recognition, as practiced in the context of the Recognizing Textual Entailment (RTE) challenges, is the task of determining whether a text entails a hypothesis in a common sense way. Two text-hypothesis entailment pairs from the RTE-2 challenge (Bar-Haim et al. 2006) are given in examples (9-10). In both of them, entailment holds.

(9) T.: Mr. Fitzgerald revealed he was one of several top officials who told Mr. Libby in June 2003 that Valerie Plame, **wife** of the former ambassador Joseph Wilson, worked for the CIA.

H.: Valerie Plame is **married** to Joseph Wilson.

(10) T.: An avalanche has struck a popular skiing resort in Austria, **killing** at least 11 people.

H.: Humans **died** in an avalanche.

Existing systems that tackle the entailment task use various techniques for judging entailment, including measuring lexical overlap, shallow syntactic parsing, and the use of WordNet relations. Another kind of approach consists in using shallow semantic representations that abstract away from semantically irrelevant variations. FrameNet, by grouping word senses into frames, provides this sort of abstraction. The goal in using FrameNet in the context of the RTE-task is thus to derive semantic normalization from it. For instance, in an entailment pair like (9), an ideal entailment recognition system would first correctly annotate text and hypothesis with a well-performing shallow semantic parser and then take the fact that the noun *wife* in the text and the adjective *married* in the hypothesis both belong to the *Personal_relationship* frame as a piece of evidence that the text entails the hypothesis.

Burchardt et al 2009 performed an experiment on the gold standard data of the FATE corpus (Burchardt and Pennacchiotti 2008) to see how much frame semantic information

can aid in discriminating between positive and negative entailment pairs. The FATE corpus contains manual frame semantic annotations for the 400 positive and the 400 negative text-hypothesis pairs of the RTE-2 task. A total of 4490 frame instances are annotated. Burchardt et al.’s (2009) approach consists in extracting frame-based statistical information from the positive and negative examples of the annotated corpus, respectively, and measuring the overlap of frame structures between text and hypothesis in an entailment pair. The key assumption underlying their method is that the more of the semantics of the hypothesis can be embedded into the text, the more likely it is that an entailment relation holds between text and hypothesis. Burchardt et al. measure the frame overlap between the sentences in the hypothesis and the sentences in the text for each entailment pair, where frame label overlap is defined as “the percentage of frame labels in the hypothesis which ‘match’ in the text”. As an example, for an entailment pair in which the text contains one instance each of the *Surviving*, *Likelihood*, and *Medical_conditions* frames and the hypothesis sentence contains one instance each of the *Medical_conditions* and *Recovery* frames, an overlap of 0.5 results for that pair. Finally, the overlap scores are averaged across all positive pairs and across all negative pairs, respectively. The difference between these two scores is considered to be a measure of the discriminative power contributed by using frame semantic information. Burchardt et al. (2009) concluded in their study that frame semantic information holds significant discriminative power.

5.2.1. Experimental setup

For both relation- and lexical unit-based transformations, we replicate Burchardt et al.’s experiment in order to assess whether the collapsing of word senses by the FN transformer causes the loss of information needed for the RTE task. To that end, we first automatically relabel those frame instances in the Fate gold standard corpus which should be reassigned to a different frame according to mappings produced by the two transformer setups we use. The frame annotations that were not affected by the transformation remain in the corpus unchanged. Finally, we compute average overlap scores for positive and negative entailment pairs and compute the difference between them.

5.2.2. Results

Table 4 shows our results for the original Fate corpus annotated according to FN release 1.3, for the Fate version annotated according to the relation-based transformation, FN1.3R, and for the Fate version annotated according to the lexical unit-based transformation, FN1.3LU.

	Positive pairs	Negative pairs	Difference
FN1.3	0.5525	0.4436	0.1089
FN1.3R	0.5913	0.4845	0.1068
FN1.3LU	0.5323	0.4348	0.0975

Table 4: Average frame label overlap on entailment pairs in three versions of the Fate corpus

The results show that in relation-based merging frame label overlap increases both in positive and negative entailment pairs, with the result that the difference between them remains virtually the same as for the original FrameNet data. In the case of lexical unit-based merging, frame label overlap is surprisingly lower than for the original FrameNet and the difference between positive and negative entailment pairs drop somewhat.

5.2.3. Discussion

By merging frames connected through certain relation types, we do not lose relevant information. Combined with the fact that semantic parsing performance remains the same or even improves slightly on transformed data, this suggests that some degree of coarsening FrameNet is not harmful. In the case of lexical unit-based merging, parsing performance is also unchanged or slightly better but some relevant information may be lost.

Frame label overlap is lower on the data resulting from lexical unit-based transformation. Most likely, this is due to the fact that lexical unit-based merging will typically separate some lexical units from their original frame-mates, which is in contrast to relation-based merging, where frame-mates move to a new frame together. LU-based merging thus carries the risk that some frame matches between text and hypothesis are lost.

The hope that the number of direct frame matches between hypothesis and text might increase more in positive entailment pairs than in negative pairs as a result of transforming the FrameNet data was not fulfilled. Positive entailment pairs like (10), where *kill* and *die* belong to the same frame, *Killing*, after relation-based merging, are matched by negative entailment pairs, which also acquire direct frame matches as a result of data transformation. Additional experiments with other parameter settings for the transformer are needed to see if a better grain size for our RTE application context can be found, or if there simply is an upper bound to simple frame label overlap as a predictor of entailment between text-hypothesis pairs. The results so far show that coarsening of FrameNet can improve semantic parsing accuracy, especially on frame assignment, and that it does not lead to the loss of information relevant to the RTE-task.

6. Conclusion

We have presented a tool for semi-automatically deriving customized but format-compliant versions of the FrameNet database based on frame and frame element relations. It allows users to produce FrameNet versions whose granularity is suitable for their particular applications. In our baseline evaluations so far, we have found that coarsening FrameNet somewhat does not harm parser performance nor cause the loss of information needed for the RTE task. Additional experiments are needed to assess whether the individual gains of the two modes of transformation can be combined and what the best settings are for each of them.

Acknowledgments

We would like to thank Ines Rehbein for help with the Shalmaneser experiments. This work has partly been funded by the German Research Foundation DFG (grant PI 154/9-3).

7. References

- C.F. Baker, C.J. Fillmore, and J.B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 86–90, Morristown, NJ, USA. Association for Computational Linguistics.
- R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. 2006. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.
- A. Burchardt and M. Pennacchiotti. 2008. Fate: A framenet-annotated corpus for textual entailment. In *Proceedings of LREC 2008*, Marrakech, Morocco.
- A. Burchardt, K. Erk, and A. Frank. 2005. A WordNet Detour to FrameNet. In *Proceedings of the GLDV 2005 Workshop GermaNet II*, Bonn.
- A. Burchardt, M. Pennacchiotti, S. Thater, and M. Pinkal. 2009. Assessing the Impact of Frame Semantics on Textual Entailment. *Journal of Natural Language Engineering*.
- K. Erk and S. Pado. 2006. Shalmaneser - a flexible toolbox for semantic role assignment. In *Proceedings of LREC 2006*, Genoa, Italy.
- H. Fürstenuau and M. Lapata. 2009a. Graph Alignment for Semi-Supervised Semantic Role Labeling. In *Proceedings of EMNLP 2009*, Singapore.
- H. Fürstenuau and M. Lapata. 2009b. Semi-Supervised Semantic Role Labeling. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, Athens, Greece.
- M. Kaisser and B. Webber. 2007. Question answering based on semantic roles. In *Proceedings of the ACL 2007 Deep Linguistic Processing Workshop (ACL-DLP 2007)*, Prague.
- Y. Matsubayashi, N. Okazaki, and J. Tsujii. 2009. A Comparative Study on Generalization of Semantic Roles in FrameNet. In *Proceedings of the 47th Annual Meeting of the ACL*, pages 19–27, Suntec, Singapore.
- M. McConville and M.O. Dzikovska. 2008. Using inheritance and coreness sets to improve a verb lexicon harvested from FrameNet. In *Proceedings of the Linguistic Annotation Workshop*, pages 33–40.
- M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal*, 31(1):71–106.
- I. Rehbein, J. Ruppenhofer, and C. Sporleder. 2009. Assessing the benefits of partial automatic pre-labeling for frame-semantic annotation. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)*, Suntec, Singapore. Association for Computational Linguistics.
- D. Shen and M. Lapata. 2007. Using semantic roles to improve question answering. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague.