Dialogue and Discourse 4(2) (2013) 282-324 doi: 10.5087/dad.2013.212

A Discriminative Analysis of Fine-Grained Semantic Relations including Presupposition: Annotation and Classification

Galina Tremper TREMPER@CL.UNI-HEIDELBERG.DE
Department of Computational Linguistics
Heidelberg University, Germany

Anette Frank FRANK@CL.UNI-HEIDELBERG.DE
Department of Computational Linguistics
Heidelberg University, Germany

Editors: Stefanie Dipper, Heike Zinsmeister, Bonnie Webber

Abstract

In contrast to classical lexical semantic relations between verbs, such as antonymy, synonymy or hypernymy, presupposition is a lexically triggered semantic relation that is not well covered in existing lexical resources. It is also understudied in the field of corpus-based methods of learning semantic relations. Yet, presupposition is very important for semantic and discourse analysis tasks, given the implicit information that it conveys. In this paper we present a corpus-based method for acquiring presupposition-triggering verbs along with verbal relata that express their presupposed meaning. We approach this difficult task using a discriminative classification method that jointly determines and distinguishes a broader set of inferential semantic relations between verbs.

The present paper focuses on important methodological aspects of our work: (i) a discriminative analysis of the semantic properties of the chosen set of relations, (ii) the selection of features for corpus-based classification and (iii) design decisions for the manual annotation of fine-grained semantic relations between verbs. (iv) We present the results of a practical annotation effort leading to a gold standard resource for our relation inventory, and (v) we report results for automatic classification of our target set of fine-grained semantic relations, including presupposition. We achieve a classification performance of 55% F_1-score, a 100% improvement over a best-feature baseline.

Keywords: Presupposition, entailment, question-based annotation, automatic classification.

1. Introduction

Computing lexical-semantic and discourse-level information is crucial in event-based semantic processing tasks. This is not trivial, because significant portions of content conveyed in a discourse may not be overtly realized. Consider the examples (1.a) and (1.b), where (1.a) bears a presupposition that is overtly expressed in (1.b):

(1) a. Spain won the finals of the 2010 World Cup.

b. Spain played the finals of the 2010 World Cup.

The presupposition expressed in (1.b) is implicitly encoded in (1.a), through lexical knowledge about the verb *win*, and is thus automatically understood by humans who interpret (1.a), given their linguistic knowledge about the verbs *win* and *play*. Automatically acquiring this kind of lexical semantic information is one of the objectives of the present work.

Submitted 04/12; Accepted 01/13; Published online 07/13

One reason for embedding the acquisition of presupposition-triggering verbs in a discriminative classification task is that *presupposition* needs to be carefully distinguished from other lexical relations, in particular *entailment*. The two relations are closely related, but crucially differ in specific aspects. Consider the sentence pair in (2).

(2) a. President John F. Kennedy was assassinated.

b. President John F. Kennedy died.

Sentence (2.a) logically entails (2.b). But how does this differ from the presuppositional relation between (1.a) and (1.b)? Generally speaking, *entailment* is a strictly logical implication relation holding between propositions p and q in such a way that whenever p holds true, q also holds true. *Presupposition*, by contrast, is a relation that may be perceived as holding between propositions, but is often viewed as a pragmatic relation holding between a speaker and a proposition. Crucially, *presuppositions* are what a speaker assumes to hold true as a precondition for a sentence to be true. Our focus is on presuppositions as conventional implicatures, as opposed to conversational implicatures (Levinson, 1983).

There are a variety of linguistic sources for presuppositions, including possessive pronouns, definite reference, or cleft-/wh-constructions that trigger specific presuppositions, such as possessive relations or existence. While these constitute a closed list, we are interested in lexically triggered presuppositions, mainly by verbs, that are grounded in the lexical meaning of the triggering predicates. Examples are widespread, including aspectual verbs such as *begin/start doing X – not having done X before* but most importantly general action verbs such as *win – play*, *know – learn*, *find – lose*, etc. Thus, in this work, we concentrate on a notion of presupposition that is restricted to the lexical meaning relation holding between the presupposition-triggering verb and the verbal predicate of the evoked presupposition. But then again, how to distinguish between verb pairs that characterize lexically triggered presuppositions as in (1) from pairs of verbs that license a classical entailment relation as in (2)?

The differences between *presupposition* and *entailment* can be studied using special presupposition tests (Levinson, 1983). The most compelling one, which we will use throughout, is the negation test. It shows that presupposition is preserved under negation, while entailment is not. Applied to (1) and (2), we note that (3.a), the negation of (1.a), still implies (1.b), while (3.b), the negation of (2.a), does not imply (2.b). This can be taken as evidence that *win* lexically presupposes *play*, while *assassinate* and *die* are lexical licensors for logical entailment.

(3) a. Spain didn't win the finals of the 2010 World Cup.

b. President John F. Kennedy wasn't assassinated.

The negation test not only helps us to distinguish these closely related verb relations. It also points to the distinct behavior of these relations in deriving implicit meaning from discourse, which is the main motivation underlying our work. If we encounter the verb *win* in the intended meaning *x wins the game* in some piece of discourse, we may infer *x played the game* – whether the phrase is negated or not. For a verb that stands in an entailment relation, by contrast, we need to make sure that the triggering verb is not in the scope of negation. So, *x was killed* implies that *x died*, but *x wasn't killed* does not license this inference.

Similar to entailment, presuppositions are essentially grounded in world knowledge. At the same time, they are crucial for the computation of discourse meaning and inference. This is exemplified in (4), a typical case of presupposition that introduces additional, implicit knowledge, by so-called *accommodation* behavior (van der Sandt, 1992; Geurts and Beaver, 2012). The predicate *lift* licenses the presupposition that the ban on deep sea drilling that has been lifted had previously been imposed. Because this presupposition is lexically triggered, it causes anyone unaware of this piece of world knowledge to infer that a moratorium on deep-water drilling had been *imposed* for the Gulf of Mexico some time before October 12, 2010, the publication date of the article.

(4) The Obama administration lifted its moratorium on deep-water drilling in the Gulf of Mexico Tuesday, replacing it with what Interior Secretary Ken Salazar is calling a gold standard of safety standards for operators looking to drill in water depths greater than 500 feet.[1]

It is their relevance for discourse understanding and inference that motivates capturing lexical semantic relations in computational lexicons, to make them available for lexically driven inferences in NLP applications (Frank and Pádo, 2012). Among these are the major taxonomic lexical semantic relations, such as *antonymy*, *synonymy* or *hypernymy* that are grounded in linguistic tradition (Lyons, 1977) and that form the core of lexical semantic resources such as WordNet (Fellbaum, 1998). Recent efforts in computational linguistics further aim to automatically acquire lexical relations that determine linguistically licensed inferences, such as *entailment* and other more fine-grained relations, which are not yet covered in sufficient detail and coverage in the WordNet data base.

Chklovski and Pantel (2004) were first to attempt the automatic classification of fine-grained verb semantic relations, such as *similarity*, *strength*, *antonymy*, *enablement* and *happens-before* in VerbOcean. In the present paper we aim to extend the classification of semantic relations between verbs to lexical inferences licensed by *presupposition*. To our knowledge, this has not been attempted before. We will address this task in a corpus-based discriminative classification task – by distinguishing presupposition from other semantic relations, in particular *entailment*, *temporal inclusion* and *antonymy*.

Our overall aim is to capture implicit lexical meanings conveyed by verbs, and to make this knowledge explicit for improved discourse interpretation by lexically induced inferences. This overall aim can be divided into two tasks:

Detecting and discriminating fine-grained semantic relations: We first detect and distinguish fine-grained semantic relations holding between verbs at the type level, to encode this lexical knowledge in lexical semantic resources.

Deriving implicit meaning from text: In a second step, we will apply this knowledge for the interpretation of discourse, at the context level, in order to enrich the overtly expressed content with *implicit* knowledge conveyed by presupposition, entailment, or other lexically supported semantic inferences. That is, when detecting a verb in a given piece of discourse that stands in a particular meaning relation with some other verb, we apply the learned lexical knowledge to enrich the discourse representation with this hidden meaning relation, by lexically driven inferences. Through the inferred semantic knowledge we obtain densely structured semantic representations of discourse that can improve the quality of automatic semantic and discourse processing tasks, such as information extraction, text summarization, question-answering and full-fledged textual inferencing or natural language understanding tasks.

[1] Source: The Christian Science Monitor, Oct. 12, 2010.

The present paper concentrates on the first task. We present a corpus-based method for learning semantic relations between verbs, with a special interest in detecting verbs related by or triggering presuppositions. Learning focused lexical semantic relations from corpora is a hard task. Our main strategy for approaching this task is to design features for classification that are able to discriminate presupposition from other lexical relations. A novel aspect of our work is that we employ *type-based* features that are derived from *logical-semantic properties* of the targeted lexical relations.

As it turns out, the classification we aim to perform is even difficult for humans: the complex inference patterns that characterize the differences between the semantic relations we consider are difficult to discern using classical annotation schemes. We devise a question-based annotation design that yields reliable annotation results. On the basis of the resulting annotated data set we will present first results for automatic discriminative classification of fine-grained semantic relations between verbs using alternative classification architectures.

The structure of the paper is as follows: Section 2 reviews related work. Section 3 motivates the choice of our target set of semantic relations and studies their discriminative properties. Section 4 discusses different annotation strategies and their difficulties and develops a question-based annotation scenario that yields improved annotation quality. In Section 5 we present two classification experiments and the results we obtain. We present an error analysis and compare our results to related work. Finally we summarize and present conclusions in Section 6.

2. Related Work

Semantic relation acquisition. Significant progress has been made during the last decade in automatic detection of semantic relations between pairs of words, using corpus-based methods. The majority of approaches follow the *distributional hypothesis*: semantically related words tend to occur in similar contexts (Firth, 1957). Two types of methods can be distinguished in this field.[2]

Pattern-based methods make use of specific lexico-syntactic patterns that identify individual relations, e.g., the *such as* patterns used by Hearst (1992) to detect hyponymy (*is-a*) relations between nouns. Similar techniques have been applied to detect *meronymy* relations (Girju et al., 2006). In contrast, *distributional methods* record co-occurring words in the surrounding context of a target word, and compute semantic relatedness between two target words using measures of distributional similarity such as *cosine* or *Jaccard* (Mohammad and Hirst, 2012).

The strength of pattern-based approaches is that particular relations can be identified with high precision, if effective relation-identifying patterns can be determined. Often, however, pattern-based approaches are critically lacking recall. Distributional approaches do not suffer from such coverage problems. But distributional measures of ‘similarity’ and ‘relatedness’ are in general not specific enough to permit a clear-cut distinction of individual meaning relations (Baroni and Lenci, 2011).

Pantel and Pennacchiotti (2006) propose a weakly supervised pattern-based bootstrapping algorithm, *Espresso*, that addresses the recall problem. It admits generic patterns – high-recall, yet low-precision patterns – which may refer to more than one semantic class. In conjunction with *Espresso*’s refined filtering methods, generic patterns yield high recall without loss of precision.

In our approach, we will perform semantic relation classification in a different way, using features for classification that encode more abstract *linguistic properties* of individual relation types. This way we avoid the fuzziness of distributional measures and, at the same time, compensate for the lack of discriminative surface patterns for the inferential relations we need to distinguish.

[2] See Frank and Pádo (2012) for an overview.

Acquisition of (verb) inference rules. A related strand of work aims at the automatic acquisition of inference rules. Engendered by the Recognizing Textual Entailment (RTE) challenges, the main goal is to identify inference relations holding between two pieces of text, such that one of them can be inferred from the other (Dagan et al., 2009). The notion of inference that underlies the RTE challenges is informally defined as the *most probable* inference that can be drawn from some text, relying on common human understanding of language and background knowledge.

Pekar (2008), Aharon et al. (2010), Berant et al. (2012) and Weisman et al. (2012) extract broad inferential relations between verbs, without sub-classifying them into more fine-grained relation types, such as *presupposition, entailment* or *cause*. However, knowledge about the specific inferential properties of these relations is crucial for drawing correct inferences in a given context.

Distinguishing fine-grained semantic relations between verbs. Only few attempts tried to further distinguish inferential relations between verbs.

Chklovski and Pantel (2004) performed fine-grained semantic relation classification with VerbOcean. They built on work by Lin and Pantel (2001), who proposed a distributional measure that extracts highly associated verbs. Chklovski and Pantel (2004) took Lin's semantically associated verb pairs as a starting point and applied a semi-automatic pattern-based approach for determining fine-grained semantic relation types, including *similarity* (synonyms or siblings), *strength* (synonyms or siblings, where one of the verbs expresses a more intense action), *antonymy*, *enablement* (a type of causal relation) and *happens-before*. This inventory of semantic relations is different from ours. In contrast to VerbOcean, we do not consider *synonymy* and *strength*. Also, there is no direct mapping from their entailment relations *enablement* and *happens-before* to our target relations.

Inui et al. (2005) concentrate on the acquisition of causal knowledge. They sub-classify causal relations into the four types: *cause*, *effect*, *precondition* and *means*, using the Japanese connective marker *tame* as a contextual indicator. They distinguish two types of events: actions (Act) and states of affairs (SOA). For *cause(*SOA_1, SOA_2*)* and *effect(*Act_1, SOA_2*)*, SOA_2 happens as a result of SOA_1 or Act_1, respectively. With *precond(*SOA_1, Act_2*)*, Act_2 cannot happen until SOA_1 has taken place. Finally, *means(*Act_1, Act_2*)* involves two actions sharing agents and that can be paraphrased as Act_1 *in order to* Act_2. Unlike Inui et al. (2005) we do not distinguish subclasses of causal relations, but consider them as special cases of *entailment*.

Important work on clarifying the implicative properties of verbs has been presented by Karttunen (2012). Similar to our work, he tries to divide implicative constructions into different types, but in contrast to our work, he studies the relation between the implicative verb (phrase) and its complement clause. Karttunen (2012) identifies different types of implicative signatures and classifies the verbs accordingly. For example, *refuse to* is a one-way implicative verb with the implicative signature $+-$: the entailment applies in affirmative contexts only, and consists in negating the complement clause. At present, this classification has not been automated.

Computing presuppositions. Only little work is devoted to the computational treatment of presupposition. Bos (2003) adopted the algorithm of van der Sandt (1992) for presupposition resolution. His approach is embedded in the framework of DRT (Kamp and Reyle, 1993). It requires heavy preprocessing and a lexical repository of presuppositional relations. Clausen and Manning (2009) compute presuppositions in a shallow inference framework called 'natural logic'. Their account is restricted to computing factivity presuppositions of sentence embedding verbs. In the field of corpus-based learning of semantic relations, the automatic acquisition of presupposition relations remains understudied.

3. A Corpus-based Method for Learning Semantic Relations

We present a corpus-based method for learning semantic relations between verbs with a focus on verbs involved in lexically triggered presupposition relations. In order to better capture the specific properties of presuppositional relations, we embed this task in a discriminative classification setup. As target classes we initially consider five relation types: *presupposition, entailment, temporal inclusion* (which covers *troponymy* and *proper temporal inclusion*), *antonymy* and *synonymy* that we aim to differentiate, as well as a negative class of verb pairs related by some other relation, or that do not stand in any relation at all (*other/unrelated*).[3]

3.1 Selection of Target Semantic Relations

This target set of relations is motivated by three criteria. First of all, we aim at a broad space of relation types, in order to acquire a wide spectrum of relations that bear inferential characteristics. For this reason, our selection encompasses the taxonomic relations *hypernymy/troponymy, synonymy* and *antonymy*, which have proven efficient in computational textual entailment and question-answering tasks, as well as classical non-taxonomic inference relations. Second, as our focus is on relations between verbs, the relations should be characteristic for verbs. Finally, we need to choose relation types that are sufficiently discriminative to permit automatic subclassification using corpus-based methods.

Inferential relations (between verbs). Lexical resources such as WordNet (Fellbaum, 1998) or GermaNet (Kunze and Lemnitzer, 2002) cover the core taxonomic relations *synonymy* (through the notion of synsets), *antonymy* and *hypernymy/hyponymy*. In the verbal domain, *hypernymy* corresponds to the special relation *troponymy* (for instance, *march – move*, *mutter – talk*).[4] These relations are clearly inferential: for synonymous verbs V_1 and V_2 and a proposition p_{v_1} based on V_1, we can infer p_{v_2/v_1}, i.e., the proposition p_{v_2} that results from substituting V_1 with V_2 and vice versa. Antonymy allows us to infer $\neg p_{v_2}$ from p_{v_1}. For hypernymy or troponymy, we can infer p_{v_2} from p_{v_1}, but we cannot infer p_{v_1} from p_{v_2}.

Cutting across these taxonomic relations, which apply to all major open class categories, we find inferential relations that are specific to verbs. These are based on temporal, causal, or inferential relations that are grounded in world knowledge about events: *temporal inclusion, causation*, *entailment* or *presupposition*. Temporal inclusion (*sleep – snore*) differs from troponymy in that *snoring* is not a special way of *sleeping* but merely an action that may occur *while* sleeping. *Causation* can be considered a special form of entailment that involves a physical or other external force that brings about a state of affairs: *feed – eat, kill – die* (Carter, 1976). Finally, we find a broad class of verbs that lexically *entail* or *presuppose* one another, such as *breathe – live* or *win – play*.[5] They typically do not instantiate hierarchically related concepts as in troponymy, but can be characterized as 'log-

[3] In fact, we will exclude *synonymy* later on, for reasons relating to the specific corpus-based methods we apply. Nevertheless we include it here, for the general discussion of the inferential properties of lexical-semantic relations.

[4] While WordNet (Fellbaum, 1998) makes use of the *troponymy* relation for verbs, GermaNet uses the *hypernymy* relation across the different word categories (Henrich and Hinrichs, 2010).

[5] In what follows we adopt the commonly used convention, as e.g. in Fellbaum (1998), that grounds inferential relations holding between propositions to their licensing verbs. I.e., we refer to pairs of verbs V_1 and V_2 that are able to license *entailment* or *presupposition* relations between propositions p_{v_1} and p_{v_2} as standing in a *lexical entailment* and *presupposition* relation, respectively.

ical consequences' or 'preconditions' of each other and are grounded in real-world knowledge. All of the latter relations are unidirectional, except for entailment, for which modus tollens holds.[6]

Selecting target relations for classification. Fellbaum (1998) establishes a hierarchy of inferential relations between verbs that distinguishes four types of lexical entailment: *troponymy* and *proper temporal inclusion* (which both involve a temporal inclusion relation between verbs) are distinguished from *backward presupposition* and *cause* (which do not involve temporal inclusion).[7]

This relation inventory is very fine-grained. In practice it is difficult to discriminate relation instances along the relevant criteria, such as 'external force' for *causation*, or proper temporal inclusion vs. coextensiveness, in order to discriminate *proper temporal inclusion* from *troponymy*. In fact, although Fellbaum's hierarchy distinguishes four relation types, *backward presupposition* and *proper temporal inclusion* have been grouped together (Richens, 2008).[8]

In our approach we adopt a different relation hierarchy (see Figure 1). We adopt WordNet's basic taxonomic relations *synonymy*, *antonymy* and *troponymy* (as a special class of *hypernymy* in the verbal domain). Unlike WordNet, we range *causation* with the more general *entailment* relation. Similar to WordNet we group *proper temporal inclusion* with *troponymy* as they share inferential properties, but distinguish *entailment* (inclusive of *causation*) from *presupposition* since these relations show distinct inferential behavior. The latter two classes differ from the former, as the verbs are involved in temporal sequence (precedence, overlap or succession).[9] This leaves us with five relations that span a large range of inferential relations: taxonomic and non-taxonomic, symmetric and asymmetric, that we set out to distinguish using corpus-based classification.

[6] We follow the classical definitions for *presupposition* and *entailment*, as given below:

Presupposition is defined by Strawson (1950) as follows:

> A statement A *presupposes* another statement B iff:
> (a) if A is true, then B is true; (b) if A is false, then B is true.

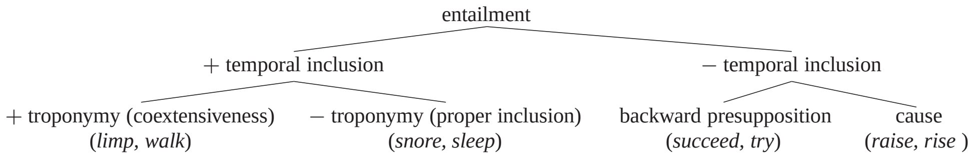
Condition (b) is known as the property of *persistence under negation* that is characteristic for presupposition. The *backward presupposition* relation in WordNet is based on this definition, and like Fellbaum (1998) we ground the presupposition relation holding between propositions to a lexical relation holding between the presupposition-triggering verb and the verbal predicate of the triggered presupposition.

Entailment, also referred to as *logical consequence*, can be defined as follows:

> A *semantically entails* B iff every situation that makes A true, makes B true. (Levinson, 1983)

Similar to presupposition we consider only lexical entailment relations holding between verbs that determine entailment between propositions A and B.

[7] Her terminology differs from the one adopted here, with 'entailment' being largely equivalent to our use of 'inferential'. The structure of the WordNet entailment hierarchy is reproduced below.



[8] By grouping *(backward) presupposition* and *cause* together as special forms of *entailment*, WordNet collapses two relation types with clearly distinct inferential properties, especially with regard to negation and cancellation (cf. Section 1 and below discussion of (5)–(7), p. 289 and Table 2, p. 290). Moreover, *causation* can be considered a special form of *entailment*, while in this taxonomy *entailment* is not represented as an individuated semantic relation type.

[9] Note that the distinction between proper temporal inclusion and cases of overlap in temporal sequence is difficult. However, we adhere to this distinction, as introduced by Fellbaum (1998), as indeed we find clear differences in the inferential properties of these two types of verb relations.

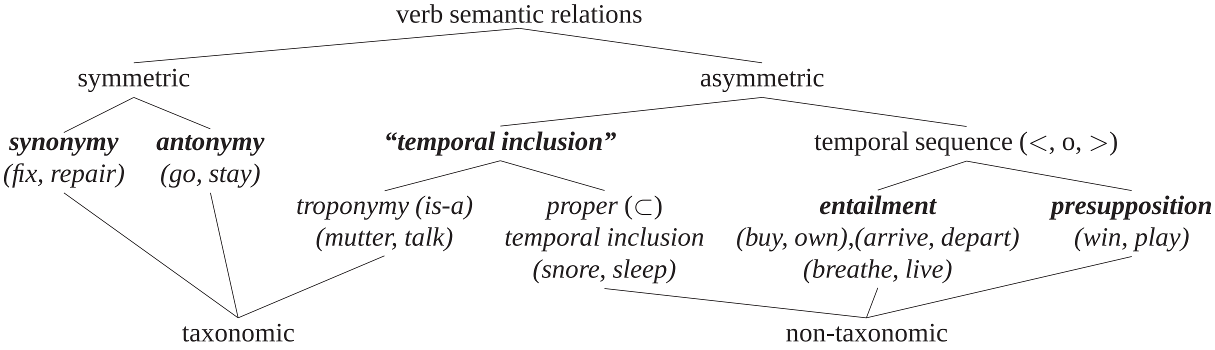


Figure 1: Hierarchy of inferential semantic relations, with selected classes printed in bold.

± Temporal	Semantic	Example	Behavior under Negation	
Sequence	Relation	(V_1, V_2)	(V_1, V_2): $I_x : p_{\pm v_1} cond\ p_{\pm v_2}$	(V_2, V_1): $I_x : p_{\pm v_2} cond\ p_{\pm v_1}$
Temporal Precedence (V_1 *prec* V_2) $V_1 < V_2$	Entailment	*(buy, own)*	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $- \Diamond\!\!\rightarrow +^{\mathbf{e}}$ I$_3$: $- \Diamond\!\!\rightarrow -$ I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	I$_1$: $+ \Diamond\!\!\rightarrow +$ I$_2$: $+ \Diamond\!\!\rightarrow -^{\mathbf{e}}$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(- \Diamond\!\!\rightarrow +)$
Temporal Succession (V_1 *succ* V_2) $V_2 < V_1$	Entailment	*(arrive, depart)*	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $- \Diamond\!\!\rightarrow +^{\mathbf{e}}$ I$_3$: $- \Diamond\!\!\rightarrow -$ I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	I$_1$: $+ \Diamond\!\!\rightarrow +$ I$_2$: $+ \Diamond\!\!\rightarrow -^{\mathbf{e}}$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(- \Diamond\!\!\rightarrow +)$
	Presupposition	*(win, play)*	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $- \Diamond\!\!\rightarrow +^{\mathbf{p}}$ I$_3$: $- \Diamond\!\!\rightarrow -^{\mathbf{c}}$ I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	I$_1$: $+ \Diamond\!\!\rightarrow +$ I$_2$: $+ \Diamond\!\!\rightarrow -$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(- \Diamond\!\!\rightarrow +)$
Temporal Overlap (V_1 o V_2)	Entailment	*(breathe, live)*	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $- \Diamond\!\!\rightarrow +^{\mathbf{e}}$ I$_3$: $- \Diamond\!\!\rightarrow -$ I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	I$_1$: $+ \Diamond\!\!\rightarrow +$ I$_2$: $+ \Diamond\!\!\rightarrow -^{\mathbf{e}}$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(- \Diamond\!\!\rightarrow +)$
− Temporal Sequence	Temporal Inclusion (Proper T.I. & Troponymy)	*(snore, sleep)* *(mutter, talk)*	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $- \Diamond\!\!\rightarrow +^{\mathbf{p}}$ I$_3$: $- \Diamond\!\!\rightarrow -^{\mathbf{c}}$ I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	I$_1$: $+ \Diamond\!\!\rightarrow +$ I$_2$: $+ \Diamond\!\!\rightarrow -$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(- \Diamond\!\!\rightarrow +)$
	Antonymy	*(love, hate)*	I$_1$:$\neg(+ \Diamond\!\!\rightarrow +)$ I$_2$: $- \Box\!\!\rightarrow +^{\mathbf{t.n.d.}}$ I$_3$:$\neg(- \Diamond\!\!\rightarrow -)^{\mathbf{t.n.d.}}$ I$_4$: $+ \Box\!\!\rightarrow -$	I$_1$: $\neg(+ \Diamond\!\!\rightarrow +)$ I$_2$: $+ \Box\!\!\rightarrow -$ I$_3$: $\neg(- \Diamond\!\!\rightarrow -)^{\mathbf{t.n.d.}}$ I$_4$: $- \Box\!\!\rightarrow +^{\mathbf{t.n.d.}}$
	Synonymy	*(fix, repair)*	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $\neg(- \Diamond\!\!\rightarrow +)$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	I$_1$: $+ \Box\!\!\rightarrow +$ I$_2$: $\neg(+ \Diamond\!\!\rightarrow -)$ I$_3$: $- \Box\!\!\rightarrow -$ I$_4$: $\neg(- \Diamond\!\!\rightarrow +)$

Table 1: Inferential properties of verb relation types. $+/-$: positive/negative polarity of V_1/V_2. $\mathbf{p}$ indicates *Persistence under Negation*; $\mathbf{c}$: *Cancellation*; $\mathbf{e}$: *Exception*; $\mathbf{t.n.d}$: *Tertium non datur*.

Inferential properties. Table 1 details the inferential properties we find with instances of verb pairs instantiating the chosen relation types. These properties will establish important criteria for the automatic classification of verb relations into the target classes.

We discriminate verb pairs (V_1,V_2) along two dimensions: their *temporal sequence properties*, in terms of the typical temporal relation holding between corresponding events (or no such relation), and their *inferential behavior*, especially with regard to their *behavior under negation*. Inferences that are found valid for the different subclasses are evaluated for both directions (i.e., with V_1 or V_2 as trigger verb) and are specified using modal conditional statements relating propositions involving the related verbs. We make use of epistemic conditionals for characterizing the inferential properties for different combinations of verb polarities, as the decisions for classification made by human annotators are best guided in terms of epistemic modal reasoning. In judging inferential patterns for related verb pairs, subjects consider whether possible situations that support the truth of an event referred to by V_1 will also support the truth of an event referred to by V_2.

For each relation type we consider four inferential patterns (I_1 to I_4) using positive ($+$) and negative ($-$) polarity of the related verbs.[10] An (epistemic) conditional that *necessarily holds true* ($p_{v_1} \;\Box\!\!\rightarrow\; p_{v_2}$) corresponds to *the valid inference* that whenever p_{v_1} is true in an (epistemically) accessible world w, p_{v_2} holds true in w. The weaker *existential reading* ($p_{v_1} \Diamond\!\!\rightarrow p_{v_2}$) is true if there is at least one (epistemically) accessible world w where p_{v_1} is true that also supports the truth of p_{v_2}. That is, we can conclude from p_{v_1} that p_{v_2} may hold true or not: $p_{v_2} \vee \neg p_{v_2}$. $\neg(p_{v_1} \;\Diamond\!\!\rightarrow\; p_{v_2})$ represents a *negative inference*, i.e., we cannot conclude p_{v_2} from p_{v_1}.

Table 1 shows a clear contrast between symmetric and asymmetric relations. The *symmetric relations synonymy* and *antonymy* show symmetric inference patterns when applying forwards and backwards inferences ($I_x : p_{v_1} cond\; p_{v_2}$ vs. $I_x : p_{v_2} cond\; p_{v_1}$). For both relation types, the inferences reflect the core logical properties of the respective relations, allowing us to infer p_{v_2} from p_{v_1} for synonymy and $\neg p_{v_2}$ from p_{v_1} for antonymy (with obvious variations for different polarities).[11]

The *asymmetric relations* (*presupposition, entailment, temporal inclusion*) all pattern alike in terms of the forwards and backwards inferences I_1 and I_4, which allow us to infer p_{v_2} from p_{v_1} in forward direction (with I_4 the corollary of I_1 in the same direction) and $\neg p_{v_1}$ from $\neg p_{v_2}$ in backward direction. In forward direction, all asymmetric relation types permit us to conclude $p_{v_2} \vee \neg p_{v_2}$ from $\neg p_{v_1}$, yet it is the inference types I_2 and I_3 that mark the core of their differences.

The inferential patterns I_2 and I_3, while superficially similar in forward direction, strictly divide *entailment* (E) (in all possible ways of temporal sequencing) from *presupposition* (P) and *temporal inclusion* (T), in that for *entailment*, applying common sense reasoning, we can infer $\neg p_{v_2}$ from $\neg p_{v_1}$ as the 'normal course of things', while for *presupposition* and *temporal inclusion* we can in general conclude p_{v_2} from $\neg p_{v_1}$, in line with the well-known inferential property of presuppositions that 'survive under negation' (Levinson, 1983). That is, the corresponding inferences I_2 for *entailment* and I_3 for *presupposition* and *temporal inclusion* represent exceptional cases for *entailment*, and cancellation of presuppositions in the case of *presupposition* and *temporal inclusion*.[12]

[10] The conditional statements used in Table 1 to characterize valid inferences serve expository purposes only. We follow the definition of conditionals using a standard definition of epistemic accessibility (see e.g. Gamut (1991)).

[11] Note that for antonymy we adopt an idealized situation of 'tertium non datur', that is, we only consider antonyms that realize the extreme ends of a scale, and ignore any intermediate values, such as *being indifferent*, for *love* and *hate*. This assumption affects the inference patterns with negative antecedents for antonymy.

[12] I_3, in forward direction, with $\neg V_1$ as a trigger verb for entailment, represents a typical form of abductive inference that is subject to cancellation (similar to I_3 for presupposition). Karttunen (2012), following Geis and Zwicky (1971), calls such non-monotonic inferences 'invited inferences'.

This can be shown by applying a number of paraphrase tests to verb pairs for the various relations, as illustrated in (5) to (7). The paraphrase pattern in (5) shows that p_{v_2} can be consistent with p_{v_1}, but it does not discriminate the underlying differences between the relation types, nor does (6), which is designed to test for 'persistence under negation' as is typical for presuppositions.

(5) You don't/didn't V_1 but you (have) V_2.[13]

(6) You don't/didn't V_1, and this is because you didn't V_2 in the first place.[14]

However, (7), which explicitly refers to exceptional situations that do not correspond to the 'normal course of events', clearly establishes that *entailment* relations are subject to exceptional conditions that can make the universal conditional fail (7.d–f), while for (7.a–c) the oddity of 'exception catching paraphrases' corroborates the behavior of *presupposition* and *temporal inclusion* as being persistent under negation in their default interpretation. It is only by explicit cancellation, as in (6), that $\neg p_{v_2}$ can be inferred from $\neg p_{v_1}$.

(7)a.–c.# You didn't *win/snore/mutter*, so you didn't *play/sleep/talk* or you might have *played/slept/talked* but something exceptional happened so that you didn't *win/snore/mutter*. (P, $V_2 < V_1$; T, $V_1 \subset V_2$; T, $V_1 \subset V_2$)

d. You didn't *arrive*, so you didn't *depart* or you might have *departed* but something exceptional happened so that you didn't *arrive*. (E, $V_2 < V_1$)

e. You didn't *buy* it, so you don't *own* it or you might *own* it but something exceptional is the case so that you didn't *buy* it. (E, $V_1 < V_2$)

f. He doesn't *breathe*, so he doesn't *live* / isn't *alive* or he might *live* / be *alive* and something exceptional is the case so that he doesn't *breathe*. (E, V_1 o V_2)

These differences are recorded in Table 1 by marking forwards inferences under negation (I_2) as subject to 'exceptions' (**e**) for all *entailment* relation types (with I_2, in backward direction, as its inverse). In contrast, I_2 is marked as the default inference (**p**: 'persistence under negation')

[13] Paraphrase instances for *presupposition* (P), *entailment* (E) and *temporal inclusion* (T):

(i) You didn't *win*, but you have *played*. (P)
(ii) You didn't *snore*, but you have *slept*. (T)
(iii) You didn't *mutter*, but you have *talked*. (T)
(iv) You didn't *arrive*, but you have *departed*. (E)
(v) You didn't *buy* it, but you *own* it. (E)
(vi) He doesn't *breathe*, but he (still) *lives* / *is alive*. (E)

[14] Example (v) is slightly anomalous, but this is not specific to the *entailment* relation, but rather due to temporal sequence properties, with V_2 following V_1, which does not conform to this specific pattern.

(i) You didn't *win*, and this is because you didn't *play* in the first place. (P)
(ii) You didn't *snore*, and this is because you didn't *sleep* in the first place. (T)
(iii) You didn't *mutter*, and this is because you didn't *talk* in the first place. (T)
(iv) You didn't *arrive*, and this is because you didn't *depart* in the first place. (E)
(v) # You didn't *buy* it, and this is because you didn't *own* it in the first place. (E)
(vi) He doesn't *breathe*, and this is because he doesn't *live* / isn't *alive* in the first place. (E)

		Inference patterns (V_1,V_2)	
Relation	Temp.Rel (V_1,V_2)	I$_x$: p$_{\pm v_1}$ *op* $p_{\pm v_2}$	Example
		I$_1$: $+ \Box\!\!\rightarrow +$	*I buy – I own*
Entailment	V_1 ($<$,o,$>$) V_2	I$_2$: $- \Diamond\!\!\rightarrow +^{\textbf{exception}}$	*I don't buy, but I (still) own*
(*buy, own*)		I$_3$: $- \Diamond\!\!\rightarrow -$	*I don't buy, so I (normally) don't own*
		I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	
Presupposition	$V_2 < V_1$	I$_1$: $+ \Box\!\!\rightarrow +$	*I win – I played*
(*win, play*)		I$_2$: $- \Diamond\!\!\rightarrow +^{\textbf{persistence}}$	*I didn't win but/when I played*
Temp. Inclusion		I$_3$: $- \Diamond\!\!\rightarrow -^{\textbf{cancellation}}$	*I didn't win – because I didn't play*
(*snore, sleep*)	$V_1 \subset$ / *is-a* V_2	I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	
		I$_1$:$\neg(+ \Diamond\!\!\rightarrow +)$	
Antonymy	no temp. seq.	I$_2$: $- \Box\!\!\rightarrow +^{\textbf{tertium n.d.}}$	*you don't love – you hate*
(love, hate)		I$_3$:$\neg(- \Diamond\!\!\rightarrow -)^{\textbf{tertium n.d.}}$	
		I$_4$: $+ \Box\!\!\rightarrow$	*you love – you don't hate*
		I$_1$: $+ \Box\!\!\rightarrow +$	*I fix – I repair*
Synonymy	no temp. seq.	I$_2$: $\neg(- \Diamond\!\!\rightarrow +)$	
(fix, repair)		I$_3$: $- \Box\!\!\rightarrow -$	*I don't fix – I don't repair*
		I$_4$: $\neg(+ \Diamond\!\!\rightarrow -)$	

Table 2: Inference patterns and paraphrases for the different relation types.

for *presupposition*, and similarly for both relation subtypes of *temporal inclusion*: proper temporal inclusion (*snore, sleep*) and *troponymy* (*mutter, talk*). Conversely, I_3 represents the case of 'cancellation' (**c**) for *presupposition* and *temporal inclusion*, whereas it represents the 'normal course of events' for *entailment*.

Table 2 summarizes these outcomes, by aligning the inference patterns for the main relation types with the inference paraphrases they support as 'normal' or 'invited' inferences, or as inferences that must be marked as exceptions.

3.2 Discriminating Properties of Semantic Relations between Verbs

As can be seen from this analysis, the inferential properties of the chosen set of relations are complex and difficult to distinguish. However, their inferential properties go along with two dimensions: temporal sequence properties on the one hand and behavior with regard to negation on the other.

Temporal sequence. We observe that the taxonomic lexical semantic relations *antonymy*, *synonymy* and *temporal inclusion* typically do not involve a temporal order. In contrast, *presupposition* relations between verbs do involve a temporal sequence. The event that is presupposed, being considered as a precondition, typically precedes the event that triggers the presupposition. The verbs which stand in an *entailment* relation may or may not involve a temporal succession: the overtly realized verb can precede or succeed the entailed verb, but we also find events that are temporally overlapping, such as *live / be alive* and *breath*.

Negation. Another important aspect is the behavior of the different semantic relations under negation. *Presupposition* and *temporal inclusion* are preserved under negation. This distinguishes them from *entailment* and *synonymy* which do not persist under negation.

		Behavior under Negation			
		(V_1, V_2)	$(\neg V_1, V_2)$	$(\neg V_1, \neg V_2)$	$(V_1, \neg V_2)$
Temporal Sequence	V_1 precedes V_2	E	(E)$^{\mathbf{e}}$	E	
	V_1 succeeds V_2	E	(E)$^{\mathbf{e}}$	E	
		P	P	(P)$^{\mathbf{c}}$	
	V_1 overlaps V_2	E	(E)$^{\mathbf{e}}$	E	
No temporal sequence		T	T	(T)$^{\mathbf{c}}$	
			A		A
		S		S	

Table 3: Properties of the Semantic Relations: P(resupposition), E(ntailment), T(emporal Inclusion), A(ntonymy), S(ynonymy); e: exceptions; c: cancellation.

In fact, these temporal sequence and negation properties cross-classify and fully distinguish the selected semantic relation classes. This is schematically represented in Table 3.

The table reads as follows. We continue to use V_1 as a placeholder for the trigger verb and V_2 for the related verb.[15] For the two dimensions *behavior under negation* and *temporal sequence* we list the possible instantiations of these relation properties in terms of different combinations of negated and non-negated verb predicates and the different sequencing possibilities: V_1 (typically) temporally precedes/succeeds/overlaps with V_2, or no temporal sequence can be determined. Within the table fields we record the relation types that support the corresponding inference patterns.

For the *presupposition* verb pair *(win, play)*, for instance, the event of winning (V_1) typically temporally succeeds the event of playing (V_2). P(resupposition) therefore fills the second row. The presuppositional relation holds in case both events are asserted to hold true. P(resupposition) therefore fills the first column, marked (V_1, V_2). The event of not winning could be interpreted in two ways: its default interpretation: persistence under negation – you do not win although you've been playing ($\neg V_1, V_2$), or else cancellation – you did not win because you did not play at all $(\neg V_1, \neg V_2)$. But crucially, winning without playing $(V_1, \neg V_2)$ does not conform with the presuppositional relation between these verbs, so the respective field remains empty.

For *entailment* pairs (E) such as *(kill, die)* or *(buy, own)*, we note that y being killed entails y being dead (V_1, V_2), but if y is not killed we do in general not conclude that y is dead $(\neg V_1, V_2)$ – unless by considering other possible causes that may not be considered relevant in the situation at hand. Thus, if y is not killed, we assume as default interpretation that (under normal circumstances) y is not dead (again – unless from some other cause) $(\neg V_1, \neg V_2)$.[16]

Both cancellation for *presupposition* (**c**) and exceptional cases for inference under negated antecedents for *entailment* (**e**) are thus marked as exceptional inference patterns (indicated by brackets) that we do not assume to find frequently realized in corpus instances.

[15] For the symmetric relations *antonymy* and *synonymy* there is no distinguished trigger verb.

[16] This assumption is debatable, as only the inverse relation $(\neg V_2, \neg V_1)$ is strictly entailed: if y is not dead, y has not been killed. However, as discussed above, we include this case as a typical form of abductive inference that is subject to cancellation as is presupposition whenever we encounter $\neg V_1$ as a trigger verb for entailment. Note that nothing hinges on this assumption regarding the discriminative power of negation properties, as entailment differs from presupposition regarding persistence under negation.

By examining these temporal and negation properties encoded in Table 3, we find that they can be used to discriminate the considered semantic relation types:

(i) *Presupposition* and *entailment* (whether or not temporally related) are distinguished on the basis of persistence under negation, which holds for *presupposition* only. The same holds for *temporal inclusion* vs. *entailment*.
(ii) *Temporal inclusion* and *presupposition* behave alike regarding negation properties, but can be distinguished in terms of temporal sequencing properties.
(iii) *Entailment* between overlapping events is difficult to distinguish from *(proper) temporal inclusion* solely on the basis of temporal properties. But due to their inferential behavior under negation, they can be clearly distinguished.
(iv) *Antonymy* clearly differs from *entailment* and *presupposition* with respect to both properties, and from *temporal inclusion*, regarding negation properties.
(v) Finally, *antonymy* and *synonymy* are opposites to each other regarding negation properties.

According to this analysis, the observed temporal and negation properties could be used to discriminate four of the five semantic relation types. *Synonymy* and *entailment* are difficult to distinguish in cases where *entailment* does not involve a temporal sequence. However, as will become clear below, in our corpus-based classification approach, we will not be able to detect verb pair candidates for the *synonymy* relation. Hence, we exclude this relation type for independent reasons and range it under the class *unrelated*. The remaining four relation types that will be subject to classification: *presupposition*, *entailment*, *temporal inclusion* and *antonymy* will be distinguished from a fifth class of unrelated verb pairs – which will include synonymous verbs, in case they (accidentally) are found to co-occur in corpus instances.

3.3 Automatic Classification of Fine-grained Semantic Relations

We pursue a *corpus-based* supervised classification approach to automatically detect and distinguish candidate verb pairs, given as types, as pertaining to one of our target semantic relation types. To this end, we exploit the insights gained from the above analysis that yielded discriminating properties of these semantic relation types on the basis of *temporal sequence* and *negation properties*. In addition, we will employ a third dimension of contextual *relatedness*, which records surface-level contextual relatedness properties of these semantic relations, using indicators such as embedding or coordinating conjunctions. These relatedness features will be utilized to distinguish semantically related from *unrelated* verb pairs, as we expect their contextual relatedness properties to be more diverse compared to semantically related verb pairs. Moreover, contextual relatedness properties can be useful in cases where temporal or negation properties are difficult.

Selecting informative 'contiguous' corpus samples. For this approach we collect corpus samples of verb pairs co-occurring in *single* sentences. Even though co-occurrence in a single sentence bears high potential for the verbs being realized in a close syntagmatic relationship, this is not necessarily so. We therefore design a set of features that can be indicative of a close syntagmatic relationship between co-occurring verbs. We will refer to these features as *contiguity features*.[17]

[17] Typical configurations of 'contiguously related' verbs are illustrated in (i).

(i.a) *Replying* to the toast [..], Dr Julia King *said* how privileged the Faculty was to have two active alumni associations.
(i.b) You can *send* us your comments by simply *clicking* on this email.
(i.c) This allows you to *connect* and *disconnect* easily.

On the basis of a corpus study, we identified properties that can be indicative for contiguously related verbs in context: the distance between verbs, their occurrence in specific grammatical configurations as indicated by dependency relations or conjunctions, and co-referential binding of the arguments of both verbs. These features will be employed for detecting contextual contiguity of verb pairs in specific contexts, and used to select context samples for classification that are informative for sub-classifying the semantic relations – including the *unrelated* class (see Section 5.3.2).

Detecting type-based features for classification. Our classification aims at assigning relation classes to *verb pair types*, and thus the feature vectors employed for classification must be defined accordingly at the type level. The temporal and negation properties we established as being discriminative for the chosen set of relations are equally *type-based*. That is, they express properties we can identify in individual context samples, but not necessarily in all of them. In a corpus-based approach, we need to capture such *type-based* properties on the basis of individual classifications at the level of corpus samples, by observing and generalizing the information found with individual corpus samples. For our main classification features, this will be obtained in the following ways.[18]

In order to predict **temporal sequence** properties as a type-level feature, we detect the temporal relation holding between individual verb pair occurrences and compute the most prevalent temporal relation type for a given verb pair on the basis of these classifications, by applying an association measure such as point-wise mutual information (PMI).

For determining the **behavior of inference under negation** we need to detect instances of all possible verb polarity combinations $\langle \pm V_1, \pm V_2 \rangle$ for different verb pairs in context. That is, we extract the information whether both verbs have positive/negative polarity, or whether the first verb has positive/negative polarity and the second verb has negative/positive polarity.

From this token-level information we compute the probability for each *polarity combination* for any given verb pair. The obtained probabilities can be mapped to the negation properties of relations as displayed in Table 3, where low probability of a polarity combination corresponds to unavailable or exceptional cases, and high probability manifests attested inference possibilities, under the respective relation.

In order to obtain type-based **relatedness** features, we raise two *contiguity* features to the type level: verb distance and relating conjunctions. Information about the average distance between verbs is crucial for distinguishing related and unrelated verb pair types. The distribution of conjunctions relating certain verb pairs can contribute indicative information for distinguishing specific semantic relations (e.g., *antonymy* or *temporal inclusion*), or may indicate that the verbs are (probably) unrelated. Finally, we measure the association between specific verb pairs on the basis of co-occurrence information manifested in a corpus, using PMI as association measure and use its strength as a type-based relatedness feature.

Supervised classification using manually labeled verb pairs (at the type level). We are going to perform supervised type-based classification using type-based feature vectors. That is, we need a training set of verb pairs annotated with the appropriate semantic relation (or the class *unrelated*)

[18] Detailed description of the features employed for classification is given in Section 5.2.

on the type level, i.e., for verb pairs out of context, and accordingly, we need a gold standard data set of unseen annotated verb pairs[19] that can be used for testing.

Features for the type-based classification will be acquired for each verb pair in the training set, and similarly for the test set, using evidence gained from corpus sentences involving verb pairs that have been determined as being *contiguously* related. The features indicating the respective relation properties are acquired from the corpus samples and raised to the type level, as described above. In our experiments, the corpus samples will be drawn from a large web-based corpus, the ukWaC corpus (Baroni et al., 2009). At this step we excluded the synonymy relation, as even in such a large corpus, synonymous verbs usually do not occur contiguously in a single sentence.

Establishing annotated training and testing data sets. In order to build appropriate training and testing data sets, we cannot make use of existing resources such as WordNet or VerbOcean, as they assume different inventories of semantic relations (see Section 3.1). We thus designed an annotation task for our target relation set, to construct training and testing data for the classification.

4. Challenges of Annotation

Annotating semantic relations, especially the relations *presupposition* and *entailment*, is a difficult task because of the subtlety of the tests and the involved decisions. In order to obtain reliable annotations it is important to define the task in an easy and accessible way and to give clear instructions to the annotators.

For an initial annotation study we randomly selected a small sample of 100 verb pairs for annotation. A further set of 250 verb pairs were annotated in a revised, question-based annotation setup. The resulting annotated data sets were used as development and gold standard test sets, respectively, for evaluating automatic semantic relation classification in Section 5. The verb pair candidates for annotation were chosen from the DIRT collection (Lin and Pantel, 2001), a collection of automatically acquired semantically related verbs (see Section 2, p. 284).

4.1 Initial Annotation Strategies

As a first take, we formulated two complementary annotation tasks: one was applied to verb pairs given as types out of context (*type-based annotation*) and another was applied to verb pairs presented in context (*token-based annotation*). We analyzed the difficulty of annotation in the respective annotation setups and examined to what degree these results correlate. In order to analyze the difficulty of annotation we gave each task to two annotators and computed the inter-annotator agreement between them.[20]

4.1.1 TYPE-BASED ANNOTATION

In this setup the verb pairs were presented to the annotators without context. Since some verbs can have more than one meaning and consequently verbs in a given verb pair can stand in more than one semantic relation, the annotators were allowed to assign more than one relation to each verb pair.

[19] We restrict the notion of ‘gold standard’ data set to the subset of manually annotated verb pairs that we use for testing.

[20] The annotators are trained computational linguistics students. They are native speakers of German with a high level of proficiency in English. The pairs of annotators which took part in the different annotation tasks are not always the same. Only one student has taken part in both tasks and her annotations were taken to analyze the correlation between the different annotations.

Semantic Relation	Pattern	Example	*Substitution in pattern*
Presupposition	V_1 presupposes V_2, *not* V_1 presupposes V_2	*win – play*	*winning* presupposes *playing* *not winning* presupposes *playing*
Entailment	V_1 implies V_2, *not* V_1 doesn't imply V_2	*kill – die*	*killing* implies *dying* *not killing* doesn't imply *dying*
Temporal Inclusion	V_1 happens during V_2 or V_1 is a special form of V_2	*snore – sleep* *mutter – talk*	*snoring* happens during *sleeping* *muttering* is a special form of *talking*
Antonymy	either V_1 or V_2, V_1 is the opposite of V_2	*go – stay*	either *going* or *staying* *going* is the opposite of *staying*
Other/unrelated	none of the above	*jump – sing*	

Table 4: Semantic Relations and Inference Patterns for Annotation.

To support the annotators in their decisions, we provided them with a couple of inference patterns and examples for each semantic relation. This is shown in Table 4.

The inter-annotator agreement (IAA) for this task was 63% corresponding to a Kappa[21] value of $\mathcal{K}$ = 0.47. This can be taken as an indication of high difficulty when annotation of these semantic relations is performed out of context.

4.1.2 TOKEN-BASED ANNOTATION

In a complementary setup, we tried to simplify the task by providing the annotators with verb pairs in their original contexts, consisting of single sentences. For this token-based annotation we chose the same 100 verb pairs and randomly selected 5 to 10 contexts for each of them (there were 877 contexts overall). In contrast to type-based annotation, we only accepted a single relation label for a given verb pair.

The inter-annotator agreement for this task was IAA = 77.4%, corresponding to a Kappa value of $\mathcal{K}$ = 0.44. Error analysis showed that the most important problems are not due to semantic relations which are difficult to distinguish (e.g., *presupposition* and *entailment*), but rather in determining whether or not there is a specific semantic relation between two verbs in a given context, i.e., the distinction between the 'unrelated/other' in contrast to the remaining semantic relation classes.

4.1.3 TYPE-BASED VS. TOKEN-BASED ANNOTATION

We examined the correlation between type- and token-based annotations by comparing the annotations of a single annotator for both annotation tasks.[22] We chose only one annotator for this comparison, because we wanted to analyze how the decisions of one and the same annotator were affected by the different annotation setups.[23] For 62% of the verb pair types we observe an overlap of labels, 28% of the verb pair types were assigned labels on the basis of the annotations in context which were not present on the type level, or else the type level label was not assigned in context, because of the small amount of contexts for a verb pair. For 10% of verb pair types we

[21] Cohen's Kappa; see Cohen (1960).

[22] Only one annotator has taken part in both annotation tasks.

[23] Since in the initial task settings no translation of verb pairs was involved (cf. Section 4.3), it was not possible to trace such differences across annotators.

found conflicting annotations (e.g., *presupposition* and *entailment*). Thus, for the most part (62%) the type-based annotation conforms with the ground truth obtained from token-based annotation. An additional 28% of verb pairs can be considered to be potentially correct. The divergences for these verb pairs could be explained by the random procedure of context extraction which does not always return appropriate contexts. They can also be explained by the difficulty for the annotator to consider all possible verb meanings for highly ambiguous verbs in type-based annotation.

4.2 A Question-based Annotation Strategy using Prototypical Arguments

Our analysis of the two annotation setups clearly shows that both are difficult, yet in different ways. Annotation on the type level is difficult because no indication is given about the intended meaning of the verbs. Hence the annotators need to consider all possible combinations of meanings for any pairing of verbs. On the other hand, presenting the pairs in their original context does not make the decision much easier. This is because some sentences involve complex structure and interpretation difficulties, which require a lot of attention and time to annotate the individual examples. In general, the inference patterns offered to the annotators as decision criteria are rather involved, so they are sometimes difficult to check – with or without context. A general drawback of token-based annotation is that it is difficult to sample appropriate contexts for a balanced annotation set across the different relation types, and that annotation is necessarily time-consuming and expensive.

In order to render the annotation task more reliable and less time-consuming, we need an annotation strategy that includes the positive elements of both annotation strategies described above and that better supports the annotators in deciding on the applicability of the inference patterns.

Prototypical arguments in type-based annotation. One solution that captures positive aspects of type- and token-based annotation could be to have annotators consider *verb pairs with prototypical arguments* instead of offering them concrete sentences as disambiguating contexts. The argument abstractions could be represented by selectional preference classes. This offers the annotators hints on relevant readings to consider without them having to read and understand involved discourse snippets. At the same time, with a single reading of the verb in focus, the annotators do not need to consider and check pairs of verbs with multiple readings. Evidently, annotation will proceed much quicker if it can be performed at the type level, even if different interpretation variants must be considered, based on selectional preference classes.

Question-based annotation. In order to support annotators in the verification of complex inference patterns, we develop a *question scenario* to collect annotations. The idea is to guide the annotator step by step through the discriminative categorizing properties, in particular temporal sequence and behavior under negation, using a cascade of case-adapted questions tailored to the verb pairs under investigation. The questions elicit the critical pieces of information needed to sub-classify the verb pair in question, according to the properties of relations displayed in Table 3.

A set of cascaded questions guide the annotator through all relevant decision criteria, where each question elicits only three possible answers: *Yes / No / Maybe*. In general, each annotation instance will be decided by three such consecutive questions. The collected answers can be used to distinguish between the target semantic relations and thus to annotate the data.

We pursued both strategies: the use of prototypical arguments and question-based annotation, and applied them jointly in a third annotation task. Examining the annotation quality obtained, we achieve considerable improvements, with an acceptable degree of inter-annotator agreement.

verb pairs with prototypical arguments	semantic relation
miss(PERSON, PERSON) – catch(PERSON, PERSON)	UNRELATED(miss, catch)
miss(PERSON, TRAIN) – catch(PERSON, TRAIN)	ANTONYMY(miss, catch)

Table 5: Enriching verb pairs with prototypical arguments.

Expert vs. non-expert annotation. Our annotators are trained computational linguistics students. Since annotation is time-consuming and expensive, an obvious question is whether this simplified annotation setup – with annotation decisions broken down into more basic units – can make this difficult annotation task accessible for non-expert annotation. If so, we could collect larger sets of annotations using crowd-sourcing (Munro et al., 2010). We will therefore compare the annotation quality obtained from linguistic experts to non-expert annotations.

4.2.1 INTEGRATING PROTOTYPICAL ARGUMENTS IN TYPE-BASED ANNOTATION

Our analysis of problems in type-based and token-based annotation clearly showed that a general problem is the difficulty to capture verb interpretation due to the ambiguity of verbs. The classification decisions crucially depend on verb interpretation and thus need to be controlled in the annotation task. Further, we need to make sure annotators consider all relevant readings. Both aspects are difficult to control in type-based annotation. In token-based annotation, annotators are often confronted with shades of meaning influenced by the specific context, which make decisions too case-specific and erroneous.

We thus opt for a type-based annotation scheme that allows us to abstract away from concrete contexts and that at the same time allows us to control for verb ambiguity. This is achieved by offering prototypical arguments of the verbs, in terms of selectional preferences computed from corpora. The presentation of the verb pairs along with prototypical arguments helps the annotators focus on specific readings of the verbs, and thus avoid inconsistent annotations.

An example is given in Table 5 for the verb pair *miss* and *catch*. When annotating this verb pair without context, two readings of *miss* may be considered: *miss (1): feel or suffer from the lack of* and *miss (2): fail to reach or get to*. For the first reading, the annotator should determine the label *unrelated*, while for the second reading, *antonymy* would be the appropriate label.

Without control of context, the annotators could miss one or the other reading, and we cannot trace which reading motivated the provided labels. Presenting the verbs with prototypical arguments as generalizations directs the annotators to the appropriate interpretation and they can determine the corresponding label. Since we record the arguments provided with the verbs, this kind of sense discrimination is available for both the learning and the classification process. It will also be crucial for inference in context, as it allows us to restrict inference of implied verb meanings to the appropriate interpretation of the trigger verb in a given context.

For the computation of prototypical arguments of verb pairs, we apply Resnik (1996)'s approach for computing selectional preference scores for verb arguments. With this we determine preference semantic classes as prototypical arguments in *subject*, *object* and *prepositional object* function.

Computing selectional association scores for verb pairs. Resnik (1996) proposes an information-theoretic measure to compute a *selectional association score* between a predicate p_i and a semantic class c that fills an argument of p_i as given in (8).[24] He defines *selectional preference strength* $S(p_i)$ as the amount of information provided by the predicate p_i for the posterior probability of co-occurring with some argument class c, compared to its prior probability. Given this measure, he computes the *selectional association score* between a predicate and a given particular class c by its relative contribution to the predicate's overall selectional preference strength.

(8) $A(p_i, c) = \frac{P(c|p_i) \log \frac{P(c|p_i)}{P(c)}}{S(p_i)}$

with $S(p_i) = \sum_c P(c|p_i) \log \frac{P(c|p_i)}{P(c)}$

Since we are dealing with pairs of verbs, we slightly modify this measure to reflect the association of a class c with both verb predicates p_i and p_j, as stated in (9).

(9) $A(p_i, p_j, c) = \frac{P(c|p_i,p_j) \log \frac{P(c|p_i,p_j)}{P(c)}}{S(p_i,p_j)}$

with $S(p_i, p_j) = \sum_c P(c|p_i, p_j) \log \frac{P(c|p_i,p_j)}{P(c)}$

We computed selectional preference scores for all verb pair candidates offered to the annotators, using the adapted measure in (9).[25] Prototypical arguments were selected manually from the arguments with the highest scores.[26]

Controlling interpretation choices in the annotation task. Having computed prototypical (preferential) argument classes for given verb pair candidates, these can be presented to the annotators as illustrated in Table 5.

However, in a number of cases prototypical arguments are not sufficient to clearly discriminate predicate interpretations. In order to detect such cases, we asked the annotators to translate the predicates into their mother language (if possible).[27] Examples of diverging interpretations are given in Table 6, together with the labels the annotators assigned for the interpretations they perceived.

Differences in translations were inspected manually. In case of divergences of interpretation, we not only record the actual interpretations chosen by the annotators, but also let the annotators re-annotate such verb pairs using the interpretation of their companion annotator as a constraint. This way we collect annotations for a maximum number of readings.

4.2.2 QUESTION-BASED ANNOTATION FOR CLASSIFYING SEMANTIC RELATIONS

The complex inference patterns that need to be considered in order to distinguish *entailment, presupposition, temporal inclusion* and *antonymy* make the annotation difficult and error-prone. We therefore devise a question-based annotation setup that breaks down these complex annotation decisions into more basic units that are easier to decide. In a step-wise manner we elicit answers that

[24] Semantic class c is taken from a conceptional taxonomy. In our work we chose WordNet (Version 3.0) as used in the NLTK implementation `http://nltk.org`.

[25] Probabilities were estimated from Sections 1 to 3 of the parsed ukWAC corpus (Baroni et al., 2009). Parsing was performed using the Stanford Parser V1.6.4, `http://nlp.stanford.edu/software/lex-parser.shtml`.

[26] We opted for manual selection for the time being, in order not to introduce noise into the annotation process.

[27] In our experiment the annotation was done for English by native speakers of German, hence translation was to German. Translation could also be into some other language (distinct from the language of the annotation task) as long as it is the same for both annotators.

Verb pair	*review*(PERSON, MATERIAL) – *teach*(PERSON, PERSON)		
Annotator	Translation V_1	Translation V_2	Relation Assigned
A1	bewerten (critique)	unterrichten (teach)	UNRELATED(review, teach)
A2	wiederholen (reexamine)	unterrichten (teach)	TEMP. INCLUSION(review, teach)

Verb pair	*cry*(PERSON) – *be scared*(PERSON)		
Annotator	Translation V_1	Translation V_2	Relation Assigned
A1	schreien (yell)	erschrecken (be scared)	TEMP. INCLUSION(cry, be scared)
A2	weinen (weep)	sich fürchten (be afraid)	UNRELATED(cry, be scared)

Table 6: Capturing sense distinctions through translation to German.

guide the annotators towards a classification using the discriminative properties we established in Section 3: properties of *temporal sequence* and *behavior under negation*.

This question-based annotation scheme naturally extends the enhanced representation of verb pairs using prototypical arguments. In fact, it is dependent on this novel representation. Using appropriate placeholders, we generate skeleton sentences for the target predicates and their prototypical arguments. These are presented to the annotators, and help them check and decide on the different relation properties that hold for the generated phrases. This novel presentation scheme can thus be considered a compromise between the context-less type-based annotation and the context-rich token-based annotation setups examined in Section 4.1.

Our method is best illustrated using an example. Figure 2 displays questions and answer possibilities for annotating the verb pair *learn – speak*. Using Resnik's selectional association scores, we determine PERSON and LANGUAGE as prototypical argument classes for this verb pair. From these abstract representations including predicate, prototypical arguments and prepositions, we generate sample phrases, as seen in question Q_0.[28] Here we elicit translations to German for the given verbs in their typical argument context, to record the interpretations perceived by the annotators.

Question Q_1 is designed to determine the temporal order in which the events typically occur. This question is offered in two ways: by generating the two verb phrases in the respective orders with appropriate temporal conjunctions (*and then; at the same time*). These options are supplemented with the corresponding fine-grained temporal relation types of Allen (1983)'s classification.[29] We target a coarse three-way distinction *before, after* and *during* that each encompasses several of Allen's relations. This was determined sufficient for classification and necessary for annotation, given that the annotators also consider borderline cases. Using the graphical representations of these relations, we defined a mapping from Allen's relations to three coarse temporal relation classes that we offered to the annotators (cf. Appendix II).[30]

[28] In order to generate natural phrases, we substitute some abstract classes like PERSON with proper names such as *John*, or LANGUAGE with *Spanish*.

[29] The annotation interface allows easy access to an overview of the relation inventory (cf. Appendix I). We employed in particular the graphical representation of Allen's relations, which proved to be very helpful for the annotators in order to decide on the appropriate relation.

[30] Note that the coarse temporal relations 'before(X,Y)' and 'after(X,Y)' include the respective overlap conditions where Y overlaps with the preceding/following X, whereas we assign 'during(X,Y)' for all cases where X is fully

Q_0: **// Characterizing the interpretation of the events: //**
Please give a translation for the verbs *learn* and *speak* in these readings:
X: *John learns Spanish.* translation: _________
Y: *John speaks Spanish.* translation: _________

Q_1: **// Determining the temporal order of events: //**
What is the typical order of the following events?
a) *John learns Spanish and then he speaks Spanish.* **X before Y: {m, o, <}**
b) *John speaks Spanish and then he learns Spanish.* X after Y: {mi, oi, >}
c) *John learns Spanish and he speaks Spanish at the same time.* X during Y: {s, si, f, fi, d, di, =}
d) More than one order of events is possible.
e) Not sure (difficult to define)

Q_2: **// Determining negation properties: X and Y? //**
John learns Spanish. Will he speak Spanish?
a) Yes (X and Y)
b) No (X and ¬Y)
c) **Maybe** (X and Y or ¬Y) – Persistence under Negation → presupposition

Q_6: **// Determining negation properties: ¬X and Y? //**
John does not learn Spanish. Will he speak Spanish?
a) Yes (¬X and Y) → none
b) **No** (¬X and ¬Y) – Cancellation → presupposition
c) Maybe (¬X and ¬Y or Y) → none

Result: PRE(SPEAK,LEARN)

Figure 2: Annotation questions for the verb pair *learn – speak*.

The next set of questions is designed to elicit inference properties with respect to negation. The verb pairs are presented in sentence pairs consisting of a declarative statement involving the first verb and a subsequent question involving the second verb. This pair inquires whether the second sentence can be assumed to hold true given the first one is considered true.[31] In case the annotator has selected **a) X before Y**, Q_2 will be chosen as a follow-up question, querying the dependence of Y (= *speak*)'s truth on X (= *learn*) holding true: **X and Y?**. Here the annotator may chose **a) Yes: X and Y** *If you learn a language, you will (be able to) speak it.* But more realistically, he or she should choose **c) Maybe: X and Y/¬Y** *You may or may not be able to speak the language after having studied it.* If the latter option is taken, the relation will be a candidate for *presupposition* (PRE(Y=SPEAK, X=LEARN)) as answer c) establishes persistence under negation. At the same time, answer c) excludes *entailment* (ENT(X=LEARN, Y=SPEAK)).[32] Given answer c) is selected for Q_2, we further check inference regarding the negation of X. This is done in question Q_6: **¬X and Y?**.

included in Y's interval. These three coarse temporal relations are intended to correspond to the relations 'precedes', 'succeeds' and 'overlap' for temporally related events as used in Table 1, p. 287.

[31] The order in which X and Y are presented as well as their temporal inflection is dependent on the answer to question Q_1. Note further that depending on the relation being considered, X and Y may change roles in being considered as trigger verbs, which fill the first argument of the relation.

[32] This judgement is dependent on an interpretation of *learn* as a non-accomplished process, in the meaning of *study*.

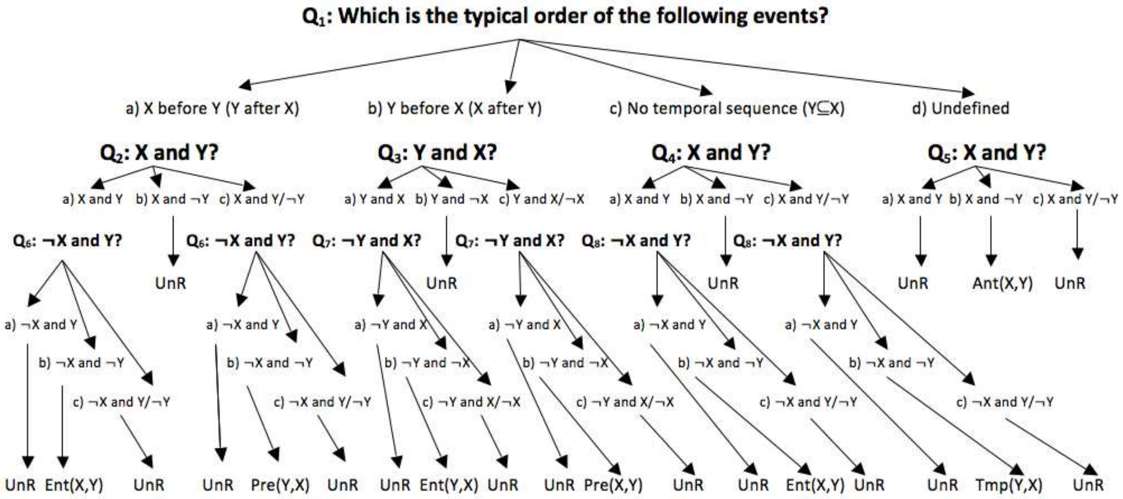


Figure 3: Decision Tree for Question-based Annotation
Pre(supposition), Ent(ailment), T(e)mp(oral Inclusion), Ant(onymy), UnR(elated).

Here, answer **b) No:** $\neg$**X and** $\neg$**Y** (i.e., *if you don't learn a language, you will not speak it*) indicates that cancellation of the presupposition *X=learn* is valid if *Y=speak* is false.

Overall, the three consecutive questions displayed in Figure 2 establish the pair *speak* – *learn* as an instance of presupposition, under an interpretation of learning as a process.

An annotation decision tree. By extending this method to the full inventory of the targeted relation types, we establish a question-based annotation scenario that takes the form of a decision tree, as displayed in Figure 3. We are able to differentiate the five relations using – in the default case – three questions per verb pair, by exploring their semantic properties, as summarized in Table 3.

The first question Q_1 clarifies the temporal sequence properties of the examined verb pair. The answer to question Q_1 also determines the order in which the consecutive tests for inference under negation are presented, e.g., Q_3 presents X and Y in a different order. This way we capture all relevant orders of verb pairs for the temporally sensitive relation types *presupposition* and *entailment*, in response to the temporal sequence properties detected in Q_1.[33] Questions Q_2 to Q_5 (all at the same level of depth) follow the very same pattern. Yet, they are dependent on the temporal properties established by the answer to question Q_1, so the answers to these questions differ in view of the relation types they may indicate. Similarly, questions Q_6 to Q_8 are structurally equivalent, but given their dependence on the previous questions and answers they will trigger case-specific conclusions as to the predicted relation type.

It should now be clear from the structure of the tree that for a verb pair such as *buy* – *own* we will obtain the classification ENT(BUY,OWN) by the following chain of questions and answers:

(10) Q_1: **which order?** a) **X before Y**
→ Q_2: **X and Y?** a) Yes: **X and Y**
→ Q_6: $\neg$**X and Y?** b) No: $\neg$**X and** $\neg$**Y**[34]

[33] For instance, the verb pair *win* and *play* cannot be classified as *presupposition* with the verbs presented as *X=win*, *Y=play*. This case is captured by response b) to question Q_1, so that the inverted verb pair relation can be tested by Q_3 (the mirror of Q_2), using inverted roles of X and Y.

Questions Q_2 and Q_3 and their follow-ups are triggered by verb pairs that involve a temporal sequence. They must be checked in both order variants to determine *entailment* and *presupposition* relations irrespective from the order in which the verb pairs are presented (see footnote 33). Question Q_4 discriminates *entailment* and *temporal inclusion* by testing persistence under negation, similar to what is done for *presupposition*. Thus, we can establish *sleep – snore* as TMP(SNORE,SLEEP) vs. *live – breath* as ENT(LIVE,BREATH). *Antonymy* is established for verbs that are not assumed to occur in sequence or concurrently, through answer d) to Q_1, which yields the value 'undefined' for temporal sequence. Here it seems sufficient to test for complementarity, brought out by answer b) **No** to Q_5: **X and Y?** for verb pairs such as *love – hate*.

Additional questions for antonymy. For some verb pairs question Q_1 yielded annotation differences depending on whether the annotators considered a syntagmatic or paradigmatic relation between the verbs. This was encountered in particular for verb pairs that qualify for both *antonymy* and *presupposition* relations, such as *open – close*, *connect – disconnect* or *accelerate – slow (down)*. Therefore, we designed an additional question for the annotators, in case we encountered that one of them had annotated a pair with *antonymy*, while the other did not. The additional questions presented to the annotator that did not annotate antonymy in the first place (here, Annotator 1) now focus explicitly on the antonymy relation. In case Annotator 1 answers both questions with **No**, the verb pair will be annotated as *antonymy*.

(11) Additional questions targeting *antonymy*:

Annotator 1 PRE(SLOW,ACCELERATE)

Annotator 2 ANT(SLOW,ACCELERATE)

$\rightarrow Q_{Ant1}$: *The car slows down. Does this car accelerate?*

$\rightarrow Q_{Ant2}$: *The car accelerates. Does this car slow down?*

Additional questions for backward entailment. In some cases the entailment relation between verbs can be symmetric, as for the pair *depart – arrive*. Such pairs should be annotated as entailments in both directions. Given the way we set up our hierarchical annotation scheme, each verb pair will only be assigned a single label. Therefore, we designed additional questions for the annotators, to identify cases of symmetric entailment. These questions take the same form as the original questions (12), but the temporal order is reversed. The answers **Yes** to the first question and **No** to the second question in (13) assign the backward entailment relation to the verb pair ENT(DEPART,ARRIVE).

(12) Standard questions targeting *entailment* generated by the annotation system:

ENT(ARRIVE,DEPART)

Q_1: **which order?** *John departs and then John arrives* (**X after Y**)

$\rightarrow$ Q_3: *John departs. Will he arrive?* a) **Yes**

$\rightarrow$ Q_7: *John doesn't depart. Will he arrive?* b) **No**

(13) Additional questions targeting *backward entailment*:

$\rightarrow Q_{Ent1}$ (= Q_2): *John arrives. Did he depart?* **Yes**

$\rightarrow Q_{Ent2}$ (= Q_6): *John doesn't arrive. Did he depart?* **No**

[34] Following our argumentation in Section 3, we ask the annotators to consider the case of 'what normally holds' in a situation if ¬X holds true and to disregard exceptional cases that are not relevant for the situation considered.

Annotation interface. In order to hide the complexity of the decision process from the annotators, this decision tree was implemented in a web-based annotation interface that presents the annotator with novel questions depending on the answers given to the previous question. The annotators were given the possibility to go back and inspect or revise the answers given to previous questions. Displays of the annotators' views for the basic question types are given in the Appendix.

Annotation quality. We evaluated the quality of annotation using this question-based annotation scheme, using 250 verb pairs selected from the DIRT collection.[35] As the novel annotation scheme is considerably simplified, we also tested it with non-expert annotators.

For the two expert annotators we obtained an inter-annotator agreement (IAA) of 72% with a Kappa value of $\mathcal{K}$ = 0.64. This is considerably higher compared to the annotation quality we obtained using standard type- or token-based annotation.[36]

This result clearly indicates that the annotation task could be dramatically simplified, with a large improvement of inter-annotator agreement. However, the decisions to be made still seem too complex for non-expert annotators: we observe poor agreement between the non-expert annotator and either of the expert annotators: IAA = 60%, $\mathcal{K}$ = 0.46 and IAA = 64%, $\mathcal{K}$ = 0.49. Thus, addressing this annotation task by crowd sourcing to non-experts does not seem to be an option in its current design.

The distribution of the semantic relations in the final annotated data set is more or less equal. *Temporal inclusion* is slightly under-represented (15%); *entailment* and *other/unrelated* are slightly over-represented (23% and 25%).[37]

5. Classification of Fine-grained Semantic Relations between Verbs

This section describes the classification architecture, employed feature sets and classification experiments for sub-classifying fine-grained semantic relations including presupposition. The performance of the classifiers is evaluated against the gold standard annotation set obtained using question-based annotation, as described in Section 4. As a reference for the subsequent description, Figure 4 summarizes the classification architectures and feature sets for the experiments described below.

5.1 Classification Method

Our aim is to acquire verb pair *types* that stand in a particular semantic relation from our selected relation inventory: *presupposition, entailment, temporal inclusion* and *antonymy* (Section 3). The lexical knowledge acquired in this way will be used to enrich textual occurrences of individually occurring trigger verbs with inferences on the basis of the learned verb relations.

For this purpose we build a classifier $\mathcal{C}_{discr}$ that automatically sub-classifies the relations holding between verb pair candidates into five classes: the four selected semantic relation types and a fifth class that captures verb pairs that stand in no or some other semantic relation not considered here. To classify the verb pairs according to our relation inventory we calculate type-based distributional features and use a supervised classification algorithm to build the model. The type-level

[35] This set is distinct from the annotation set used in Section 4.1. The annotation set produced in these initial experiments was used as development set in the classification experiments reported in Section 5.

[36] We did not perform separate evaluations of the impact of prototypical verb arguments and the break-down of annotation decisions in the question-based setting, due to the considerable annotation overhead this would have caused.

[37] This does not reflect the natural distribution of these relations, due to some amount of pre-selection for the under-represented classes.

Sample selection: $\mathcal{C}_{cntg}$: labels contiguous ([+contiguous]) corpus samples for feature extraction.

$f_{path-len}$, f_{path}: length and form of path of grammatical functions between V_1 and V_2
f_{coref}: coreference relation holding between subjects/objects of V_1 and V_2
$f_{dist-tok}$, $f_{dist-verb}$: distance between V_1 and V_2 (in tokens and verbs)
$f_{connectives}$: conjunction or direct grammatical function connecting V_1 and V_2

Type-based classification: $\mathcal{C}_{discr} : \mathcal{X} \rightarrow \mathcal{Y}$ assigns classification instances $\mathcal{X}$ consisting of pairs of verb types (V_1,V_2) one label $\mathcal{R} \in \mathcal{Y}$.

Flat: Classify instances $x \in \mathcal{X}$ into 4 core relation types plus 'U(nrelated)': $\mathcal{Y}$ = { E, P, T, A, U }.
Instance set $\mathcal{X}$: verb pair types $x \in \mathcal{X}$ (selected from DIRT (Lin and Pantel, 2001)).

Hierarchical: 1st-stage $\mathcal{C}_{rel}$: $\mathcal{C}_{rel}$ classifies all input verb pairs $x \in \mathcal{X}$ as [± related]:
[−related] if cnt([+contiguous]) < cnt([−contiguous])
& temprel = *undefined*
[+related] otherwise.

2nd-stage $\mathcal{C}_{discr}$: $\mathcal{C}_{discr}$ classifies verb pairs $x \in \mathcal{X}$ classified as [+ related] by $\mathcal{C}_{rel}$
Target classes $\mathcal{Y} \in$ { E, P, T, A }.

Feature vectors for classifier $\mathcal{C}_{discr}$ in flat (5-way) and hierarchical (4-way) classification:

Compute feature vectors $\vec{f_x} = \langle f_0, f_1, \ldots, f_n \rangle$ for all verb pair types $x \in \mathcal{X}$:

feature type	feature	flat	hier.
typical temp. rel.	f_0: $\mathcal{V} \in$ { before, during, after, undef }	✓	✓
polarity pairs	$f_1 - f_4$: $P(\langle \pm V_1, \pm V_2 \rangle \mid V_1, V_2)$	✓	✓
relatedness	f_5: average distance between V_1 and V_2 in tokens	✓	–
	f_6: PMI for V_1 and V_2 in verb pairs (V_1, V_2): $PMI(V_1, V_2)$	✓	–
	$f_7 - f_n$: cond. prob. for conjunctions c_i: $P(c_i \mid V_1, V_2)$	✓	✓

Baselines:

$\mathcal{C}_{discr}$ classifier: f_{conj}: $f_7 - f_n$: conditional probability of conjunctions c given (V_1, V_2)
$\mathcal{C}_{rel}$ classifier: $f_{connectives}$: conjunction or grammatical function relating V_1 and V_2.

Figure 4: Summary of Classification Architectures and Feature Sets.

feature vectors are calculated on the basis of a training set of corpus instances, i.e. sets of sentences involving pairs of verbs that are annotated on the type level for the relation that constitutes the classification target. The classifier learns weights for the features on the basis of the annotated training data and makes predictions for unseen verb pairs using the learned model. The performance of the classifier is tested against the set of verb relation labels defined in the gold standard data set.

Classifier definition. We define *a type-based classifier* $\mathcal{C}_{discr}$: $\mathcal{X} \rightarrow \mathcal{Y}$ that receives as input a set of instances $x \in \mathcal{X}$ of verb pair types (V_1, V_2) and a set of the feature vectors $\vec{f_x} = \langle f_1, f_2, \ldots, f_n \rangle$ calculated for any verb pair x under consideration. $\mathcal{C}$ returns one of the target class labels $R \in \mathcal{Y}$.

We experiment with two classification architectures: **flat** and **hierarchical classification**.

In the *flat classification* architecture, the classifier $\mathcal{C}_{discr}$ distinguishes all five relation types including the *unrelated* class. In *hierarchical classification* we first partition the instance set of candidate verb pairs into two classes: *related* vs. *unrelated*, with the first class covering the four selected semantic relations *P(resupposition)*, *E(ntailment)*, *T(emporal Inclusion)* and *A(ntonymy)*. In a second classification step, $\mathcal{C}_{discr}$ performs 4-way flat classification for these four relation classes, taking as input the verb pair candidates that were classified as [+ related] by the first stage classifier.

Detailed information on the setup of these architectures is given in Sections 5.3.4 and 5.3.5.

5.2 Features for Classification

The discriminative semantic relation classifier $\mathcal{C}_{discr}$ relies on the three groups of features motivated in Section 3.2: *temporal sequence*, *behavior under negation* and *contextual relatedness*.

5.2.1 TEMPORAL SEQUENCE

Our analysis of relation properties (cf. Table 3) reveals that some of our target semantic relations involve a typical temporal order, while others do not. We make this property available for discriminative classification by defining a type-based feature *typical temporal order* which records the temporal relation that can be considered typical for a given verb pair. We distinguish three basic temporal relations *before*, *after* and *during*, plus *undefined*, in case no typical temporal sequence can be determined. We obtain this information from a (token-based) temporal relation classifier.

Detecting and classifying temporal relations holding between verbs in context is a difficult task.[38] In contrast to the *TempEval* challenges (Verhagen et al., 2010), we use a coarse relation inventory that is sufficient for our purposes. Moreover, as our aim is to predict type-level temporal relation properties, we will rely on a subset of confident, i.e., reliable, token-level classifications.

A token-based temporal relation classifier. For token-based temporal relation classification we define a variety of morpho-syntactic and semantic features, including *tense*, *aspect*, *modality*, *auxiliaries*, *conjunctions*, *grammatical function paths*, *adverbial adjuncts*, *order of appearance* and *VerbNet classes (same/subsumed or different)*.[39] [40] This extends the feature set used by Chambers et al. (2007) for temporal relation classification in context.

We built a token-level temporal relation classifier that we trained on a set of manually annotated contexts, 200 contexts for each relation, using the three target temporal relations. Using the above feature set we trained a 3-way BayesNet classifier[41] for classification on the token level, with the target classes *before*, *after* and *during*. We evaluated this classifier using a set of manually annotated contexts, 20 contexts for each relation and achieved an F_1-score of 84.3% on this set.[42]

[38] See e.g. Chambers et al. (2007), Bethard and Martin (2007), Lee (2010).

[39] The VerbNet class feature is used as an indicator of temporal inclusion, in particular for the troponymy relation.

[40] We use all VerbNet classes except for *OTHER_COS-45.4*, which includes many opposite verbs, e.g. *accelerate*, *slow*.

[41] We use the BayesNet algorithm implemented in Weka (Witten and Frank, 2005).

[42] It is difficult to compare the performance of this specially designed temporal relation classifier to results reported on the TimeBank corpus, because of the different temporal relation inventories used: while we are using a coarse set of relations, the relation set used in the TempEval challenges is more fine-grained (it distinguishes 6 relations).

Predicting a type-based ‘typical’ temporal relation. We predict a type-based ‘typical’ temporal relation for any pair of verbs, relying only on confident token-level classifications (threshold 0.75). The score for each relation is computed as the association between a verb pair (V_1, V_2) and the assigned temporal relation instances in context, by applying PMI (point-wise mutual information):

$$PMI((V_1, V_2), Temp\_Rel) = \log \frac{P((V_1,V_2),Temp\_Rel)}{P((V_1,V_2))P(Temp\_Rel)}$$

For any given verb pair we choose the relation that obtains the highest PMI score. If PMI does not indicate a typical temporal relation (we set a threshold of 0.4, optimized on a held-out data set[43]), we assign the label *undefined*.

The quality of this type-level temporal relation classifier was evaluated using the answers to the first question (Q_0) of our question-based annotation scenario as a gold standard.[44] On this set it achieves an F_1-score of 73%, with balanced precision and recall at 71% and 74%, respectively.

5.2.2 NEGATION

A token-based polarity labeler. To determine the behavior under negation for given verb pairs, we first need to correctly recognize the polarity of verbs in a given context. We use a number of triggers to detect negative polarity contexts: negative particles (e.g. *not/n’t*); negative adverbs (e.g. *never*); negative adjectives (e.g. *impossible*) and negative verbs (e.g. *refuse*).[45]

In case we detect a single negation trigger for a verb in a sentence, the verb polarity is *negative*. If we find a combination of triggers (e.g. *never refuse*) and the number of triggers is even, we assign the value *positive*, if the number is odd, the verb polarity is *negative*. We also use a small set of adverbs that are able to switch a verb’s polarity in case it is negative (e.g. *badly*, etc.).

An example is given in (14). Here, the negation trigger refers to the verb *play*, but due to the combination with two negative triggers, we assign the polarity *positive*.

(14) We wanted to win the third Test as a matter of pride and **didn’t play badly** but every time New Zealand came into our 22 they scored.

To evaluate the quality of the polarity labeler, we manually annotated the polarity of 200 verbs in context.[46] On this set we achieve an F_1-score of 85%, with 84% precision and 86% recall.

Computing type-based polarity co-occurrence features. For type-based classification of verb pair polarity co-occurrences, we compute a negation vector $\vec{f}_{neg}$ with four polarity co-occurrence features for the different combinations: $\langle \pm V_1, \pm V_2 \rangle$. We compute the values of these features using the conditional probability of a given polarity co-occurrence combination for any verb pair (V_1, V_2).[47]

Another factor which influences our results positively is that we apply the classifier on contexts labeled *contiguous* in our corpus preprocessing phase. That is, we compute temporal relations only for closely co-occurring verb pairs in contiguous syntactic contexts.

[43] The held-out data consists of 100 manually annotated verb pairs.

[44] Q_0: *Which is the typical order of the following events?*

[45] We employ a manually compiled list of trigger predicates collected from various lexical resources.

[46] A subset of 100 contexts of verb pairs that were previously annotated with a semantic relation on the context level.

[47] The probability is computed using the set of verb pairs in context that are labeled [+contiguous], see Section 5.3.2.

$$\vec{f}_{neg} = \langle f_0, f_1, f_2, f_3 \rangle, \text{ with: } \begin{array}{l} f_0 = P(\langle +V_1, +V_2 \rangle | V_1, V_2) \\ f_1 = P(\langle -V_1, +V_2 \rangle | V_1, V_2) \\ f_2 = P(\langle +V_1, -V_2 \rangle | V_1, V_2) \\ f_3 = P(\langle -V_1, -V_2 \rangle | V_1, V_2) \end{array}$$

5.2.3 RELATEDNESS

Although temporal relation and negation properties can be considered discriminative for identifying our core semantic relation types, they are not sufficient for *corpus-based* classification. For example, in (15) *win* and *lose* stand in an antonymy relation, but both verbs have positive polarity. So, the evidence found in the corpus does not always correspond to the properties captured in Table 3.

(15) *Win* or *lose*, you pay nothing.

Detecting relatedness features for corpus-based classification. Thus, we include a third dimension of features that record surface-level properties of underlying linguistic properties, as in this case, where semantic opposition is not expressed by opposite polarity, but via the conjunction *or*.

Contextual relatedness features will prove particularly useful for distinguishing *antonymy* from other relation types, especially the *unrelated* class. Recall also that the discriminative relation properties that we established do not include the necessary distinction between *semantically related* vs. *unrelated* verb pairs. For the *unrelated* class, we find a broad variety of syntagmatic properties, while for the core semantic relations we find more characteristic contextual relatedness features.

Type-based relatedness features. As type-based syntagmatic *relatedness* features we employ surface-level information about *distance* and connecting *conjunctions* between verbs,[48] as well as distributional association measures, such as point-wise mutual information (PMI). These are raised to the type level in the following way (see also Figure 4):[49]

f_{dist}:	average distance between two verbs in tokens within a sentence
f_{PMI}:	PMI calculated for the two verbs in a given verb pair: $PMI(V_1, V_2)$
f_{conj}:	conditional probabilities for conjunctions c_i given specific verb pairs ($P(c_i|V_1, V_2)$)

5.3 Experiments and Results

5.3.1 DATA SETS

All candidate verb pairs that are presented to the classifier are selected from a set of *semantically related verbs* from the DIRT collection (Lin and Pantel, 2001).

Training Set. As training set we employ a small number of seed verb pairs (3 to 6 for each semantic relation) that was used in previous experiments (Tremper, 2010). We extended this data set with 30 additional verb pairs which were manually annotated by two annotators using our novel question-based annotation method (see Section 4). The overall set of 48 verb pairs yields a nearly uniform distribution of classes.

[48] We manually grouped the most informative conjunctions to a set of 21 conjunction variants, collapsing, e.g. *while/whilst, cause/because, to/in order to*. Strongest conjunctions are *or, when, if, but, by*.

[49] All values are calculated on the set of verb pairs in context that are labeled [+contiguous] (see Section 5.3.2), except for f_{PMI}, which was calculated on the basis of the full ukWaC corpus.

Test Set. As our gold standard test set we employ the annotation set consisting of 250 verb pairs that was produced using the question-based annotation setup. The distribution of relations over the 250 verb pairs is as follows: *presupposition*: 18%, *entailment*: 23%, *temporal inclusion*: 15%, *antonymy*: 19%, *other/unrelated*: 25%.

Corpus Instances. For the computation of type-based relation features we obtained corpus samples from the ukWaC corpus (Baroni et al., 2009). We extracted all sentences in which both verbs of a verb pair co-occur, considering sentences of up to 60 tokens in length. The number of contexts available for each verb pair ranges from 30 to about 500 instances.

5.3.2 PREPROCESSING: SELECTING INFORMATIVE SAMPLES FOR FEATURE EXTRACTION

To avoid noise in the computation of type-based feature vectors we need to select informative corpus instances of co-occurring verbs that stand in a close syntagmatic relation. To this end, we perform a preprocessing step that selects context samples of co-occurring verbs that are contiguously related. We designed the following set of **contiguity** features that record different types of indicators for syntagmatic relatedness of co-occurring verbs.

$f_{path-len}$, f_{path}: length and form of the path of grammatical functions relating V_1 and V_2

f_{coref}: coreference relation holding between subjects and objects of V_1 and V_2 (coreferent subjects or objects; subj coreferent w/ object; no coreference)

$f_{dist-tok}$, $f_{dist-verb}$: distance between V_1 and V_2 (in tokens and verbs)

$f_{connectives}$: subordinating or coordinating conjunction, or else direct grammatical function connecting V_1 and V_2

Using this feature set, we constructed a classifier $\mathcal{C}_{cntg}$ that labels verb pairs appearing in corpus sentences as [± contiguous]. The classifier was trained and tested on a manually annotated set of contexts involving our seed verb pairs (2343 contexts from which 90% were used for training and 10% for testing).[50] Best results were achieved using the J48 decision tree algorithm[51] (F_1-score: 79.3%).

We apply $\mathcal{C}_{cntg}$ on the set of unlabeled verb pairs in context and select all contexts that were confidently labeled as [+contiguous] (above threshold 0.75) as corpus samples for computing the feature vectors for the relation classifier $\mathcal{C}_{discr}$. Classifications obtained from the contiguity classifier are further used as a feature for the relatedness/non-relatedness classification in the hierarchical classification scenario (see Section 5.3.5 for more detail).

5.3.3 LEARNING ALGORITHMS

For our main classification task we experimented with different classification algorithms and achieved best results using BayesNet. Thus, unless noted otherwise, all results reported below were obtained using the BayesNet classifier implementation of Weka (Witten and Frank, 2005).

5.3.4 EXPERIMENT I: FLAT CLASSIFICATION

Setup. Experiment I performs classification using the *flat classification architecture*, which assigns class labels for all five relation classes including the unrelated class: $\mathcal{Y} \in \{$ *P(resupposition),*

[50] Inter-annotator agreement was 81%, with a Kappa value of 0.72.

[51] Weka implementation of the C4.5 decision tree algorithm (Witten and Frank, 2005)

Semantic Relation	Precision	Recall	F_1-score	Baseline F_1-score
Presupposition	41%	45%	43%	25%
Entailment	47%	43%	44%	25%
Temporal Inclusion	38%	47%	42%	26%
Antonymy	68%	71%	70%	47%
Other/Unrelated	54%	53%	54%	12%
All	50%	51%	51%	27%

Table 7: Results for Experiment I: Flat Classification (Baseline: best feature: f_{conj}: conjunctions).

E(ntailment), *T(emporal Inclusion)*, *A(ntonymy)*, *U(nrelated)*} (cf. Figure 4). For each verb pair in our training and test sets we compute feature vectors as described in Section 5.2.

Evaluation results. Table 7 displays the results, evaluated against the test data set. The classifier performance is compared against a baseline that uses the best single feature f_{conj}: conjunctions.

The classifier outperforms the baseline for all relation types, with balanced precision and recall. Precision is higher than recall for *entailment*. For *presupposition*, *entailment* and *antonymy* recall exceeds precision. With an overall F_1-score of 51% the classification performance is still modest, however the difference between the chosen baseline and our model is significant ($\rho < 0.05$). Note further that the average F_1-score for the more complex inferential relations (P, E, T) is lower at around 43%, while for *antonymy* it is at 70%.

5.3.5 EXPERIMENT II: HIERARCHICAL CLASSIFICATION

Setup. As an alternative to flat classification, we investigate a hierarchical architecture that first separates related from non-related verb pairs, and subsequently sub-classifies related verb pairs into the four relation classes: *P(resupposition)*, *E(ntailment)*, *T(emporal inclusion)* and *A(ntonymy)*.

A **binary classifier** $\mathcal{C}_{rel}$ separates *related from non-related verb pairs* using as criterion (i) the ratio of contexts for a given verb pair classified as [±contiguous] by the contiguity classifier in sample selection (see Section 5.3.2) and (ii) the typical temporal relation calculated by the type-based temporal relation classifier. We assign the label [$-$related] to a verb pair if the majority of contexts are annotated as [$-$contiguous] and there is no typical temporal relation for this verb pair (temprel = *undefined*).

$\mathcal{C}_{rel}$: classify all input verb pairs $x \in \mathcal{X}$ as [± related]:

[$-$related]	if count([$+$contiguous]) $<$ count([$-$contiguous]) & temprel = *undefined*
[$+$related]	otherwise.

The **second-stage discriminative relation classifier** $\mathcal{C}_{discr}$ takes as input all verb pairs classified as [$+$related] by the first-stage classifier and performs 4-way classification into the set of relation classes $\mathcal{Y}$ = { *P(resupposition)*, *E(ntailment)*, *T(emporal Inclusion)*, *A(ntonymy)*}.

Since the unrelated class has already been separated in the first classification step, the classifier does not make use of the relatedness features f_5: average distance between two verbs

1st-stage Classifier $\mathcal{C}_{rel}$: Related vs. Unrelated classification				
	Precision	Recall	F_1-score	Baseline F_1-score
Unrelated	**82%**	67%	74%	57%
Related	72%	**84%**	**77%**	54%

2nd-stage Classifier $\mathcal{C}_{discr}$: 4-way semantic relation classification (oracle input)				
	Precision	Recall	F_1-score	Baseline F_1-score
Presupposition	62%	50%	56%	30%
Entailment	53%	49%	51%	33%
Temp. Inclusion	44%	62%	52%	25%
Antonymy	76%	80%	78%	63%
All	59%	60%	59%	38%

Table 8: Exp. IIa: Individual Classifier Performance for Hierarchical Classification (with oracle). Baselines: Best features: Step 1: $f_{connectives}$: connectives; Step 2: f_{conj}: conjunctions.

and f_6: $PMI(V_1, V_2)$, as these are designed to distinguish unrelated from related verb pairs. However, the conjunction features are considered useful for discriminating the core semantic relations, and are thus included as a feature in this classification step (cf. Figure 4).

Evaluation Setup. For Experiment II we perform evaluations for both classification steps, using adapted gold standard data sets:

(i) Classifications for the *first-stage binary classifier* are evaluated against a test data set compiled from the gold standard test set used in Experiment I. It consists of the set of all unrelated verb pairs (58) and the same amount of (randomly selected) related verb pairs.

(ii) For the classifications for the *second-stage classifier* covering 4 relation classes, the test set forms the subset of the standard test data set that consists of the related verb pairs only.[52]

We report two evaluations for hierarchical classification. For both, we use best-feature baselines for the individual classifiers: the best feature f_{conj} for the discriminative relation classifier $\mathcal{C}_{discr}$ as in Experiment I, and the best feature $f_{connectives}$ for the relatedness classifier $\mathcal{C}_{rel}$.

Experiment IIa: Individual Classifier Performance. Table 8 analyzes the performance of the individual classifiers, where the second-stage classifier is based on perfect input, i.e. oracle classifications from the first-stage classifier.

The **binary classifier** $\mathcal{C}_{rel}$ obtains an F_1-score of 74% for the unrelated class, which clearly outperforms the best feature baseline by a margin of 14 points F_1-score. While the related class is recognized with higher F_1-score of 77%, we favour the results for the unrelated class, which is higher in precision (82% vs. 72%). Generally, misclassifications of the first-stage classifier impede the overall performance of the cascaded classification architecture, so while high precision is beneficial, the weaker recall (67%) could still impact the overall results.

[52] The distribution in this reduced data set is: *presupposition*: 25%, *entailment*: 31%, *temporal inclusion*: 20%, *antonymy*: 24%.

Semantic	Baseline	Flat Classification			Hierarchical Classification		
Relation	F_1-score	Precision	Recall	F_1-score	Precision	Recall	F_1-score
Presupposition	25%	41%	45%	43%	**50%**	**46%**	**48%**
Entailment	25%	**47%**	43%	44%	44%	**46%**	**45%**
Temp. Incl.	26%	38%	47%	42%	**41%**	**47%**	**44%**
Antonymy	47%	68%	71%	70%	**72%**	**74%**	**73%**
Unrelated	12%	54%	53%	54%	**68%**	**63%**	**66%**
All	27%	50%	51%	51%	**55%**	**55%**	**55%**

Table 9: Exp. IIb: Hierarchical Classification (pipeline) – Results contrasted with Flat Classification (Baseline: Best feature: f_{conj}: conjunctions).

Evaluating the **flat 4-way relation classifier $\mathcal{C}_{discr}$ on oracle classifications** we obtain an overall performance of 59% F_1-score.[53]

Experiment IIb: Full Hierarchical Classification. Table 9 presents the results for full hierarchical classification, with system input for the second-stage classifier.[54] For convenience, the results are aligned with the results obtained for flat classification in Experiment I. With an overall F_1-score of 55%, hierarchical classification significantly outperforms the baseline ($\rho < 0.05$). It also outperforms flat classification, but not significantly at a significance level of 5%. We observe performance gains for all relations, which are highest for *presupposition* (+5 points F_1-score) and *unrelated/other* (+12 points F_1-score). Again, *antonymy* performs best. Among the inferential relations, *presupposition* scores highest with 48% F_1-score and the highest precision at 50%.

5.4 Analysis of Results

5.4.1 IMPACT OF INDIVIDUAL FEATURES

We measured the impact of individual feature classes on the results, using ablation testing for different feature groups (cf. Figure 4):[55] *negation* features, *temporal sequence* features and *relatedness* features. As only the *conjunctions* feature was used in both settings, this was the only relatedness feature we omitted. The outcome, displayed in Table 10, nicely underlines the observations made in our analysis of the relation properties.

The results[56] show that temporal sequence properties are the most important feature for *entailment, presupposition* and *temporal inclusion*, whereas for *antonymy* and the *unrelated/other* class the *conjunctions* feature has the strongest effect. Eliminating conjunctions causes an overall drop to 30% (35%) F_1-score, for *antonymy* even to 15% (14%). Eliminating the temporal relation features incurs a drop to 32% (34%) with the biggest loss for *temporal inclusion*: -12 (-11) points F_1-score. Eliminating the negation features shows only a small impact of about 3–5 points in F_1-score.

[53] These figures cannot be directly compared to the flat classification results of Experiment I, which were computed over 5 classes (Table 7). However, the overall tendencies are similar.

[54] To enhance precision, we relied on the classifications for the unrelated class as input for the second-stage classifier $\mathcal{C}_{discr}$.

[55] For hierarchical classification we performed the ablation testing only for the second-stage classifier.

[56] In ablation testing, lower results indicate higher importance of the feature (group) that has been omitted.

Semantic	Exp I: Flat Classification				Exp IIb: Hierarchical Classification			
Relation	All	w/o Neg	w/o Tmp	w/o Conj	All	w/o Neg	w/o Tmp	w/o Conj
Presupposition	43%	37%	**24%**	35%	48%	41%	**22%**	34%
Entailment	44%	41%	**14%**	28%	45%	43%	**14%**	25%
Temp. Incl.	42%	42%	**12%**	38%	44%	43%	**11%**	36%
Antonymy	70%	64%	64%	**15%**	73%	68%	59%	**14%**
Other/Unrelated	54%	47%	45%	**35%**				
All Relations	51%	46%	32%	**30%**	55%	52%	34%	**35%**

Table 10: Ablation Testing: F_1-score results using different feature sets (Exp I & IIb).

Verb pair (w/ prototypical arguments)	Gold Standard	Flat Classification	Hierarchical Classification
abandon(person, do sth) – try(person, to do sth)	Pre	Pre	Pre
fly(plane) – land(plane)	Ent	Ent	Ent
multiply(person, numbers) – calculate(person, solution)	Tmp	Ent	Ent
cry(person) – laugh(person)	Ant	Ant	Ant
enter(person, house) – open(person, door)	Pre	Ent	UnR
boil(water) – evaporate(water)	Ent	UnR	UnR
steal(product) – take(product)	Tmp	Ant	Ant

Table 11: Examples of correct and wrong classifications.

Although the weakest feature type, with overall 5 points loss in F_1-score, the negation features clearly contribute to overall performance. Interestingly, they have the strongest effect for *presupposition*, with a drop of 6–7 points in F_1-score. This clearly reflects the specific negation properties found with presupposition. This analysis corroborates that while the negation properties are very important for language understanding and logical inference, and proved effective as a guide for human annotation, a corpus-based classification approach needs to complement its effects, as human language often resorts to other means for expressing negative polarity, such as the use of conjunctions (*or*, *whereas*, etc.), or does not make it explicit at all.

5.4.2 CLASSIFICATION EXAMPLES AND DIVERGENCES

Table 11 displays examples of correct and wrong classifications for both architectures. The verb pairs are given with the prototypical arguments that were used for the gold standard annotation.

5.4.3 ERROR ANALYSIS

The most frequent errors we observe (especially for the flat architecture) are misclassifications between related and unrelated verb pairs and between *presupposition* and *entailment*. This points to weaknesses of contiguity features used in the contiguous sample selection step and of negation features used for the main classification. We also notice that *entailment* is often misclassified as *temporal inclusion*. Misclassifications between *presupposition* and *temporal inclusion* are rare compared to other relations. This indicates that the temporal sequence features are effective.

As further major error sources we identified problems with verb ambiguity and coreference resolution. Both of them affect the detection of semantic relations as being related vs. unrelated.[57]

Finally, we identified errors in selecting contexts from the ukWaC corpus, which are used for computing the distributional features for the main classification. Inspection of a small section of corpus samples shows that erroneous annotations of nouns or adjectives as verbs cause errors in the computation of the type-based feature vectors. We have solved the problem of erroneous annotations of adjectives as verbs by double checking the dependencies between verbs and nouns,[58] but we still need to address the problem of erroneous annotations of nouns as verbs.

Regarding classification architectures, hierarchical classification outperforms flat classification for all relation types, and especially for the *unrelated* class. Thus, the first-stage classifier that separates related from unrelated verbs implements a strong filter. The pipeline architecture still suffers from a performance loss due to misclassifications of the first-stage classifier. This problem can be addressed in future work by using a joint classification approach.

5.5 Comparison to Related Work

Related work on semantic relation classification differs from our approach in a variety of respects (see Section 2). Nevertheless we compare our results to what could be achieved there, to give an idea about the state of the art on comparable and related tasks.

Closest to our work is VerbOcean. Chklovski and Pantel (2004) apply a semi-automatic pattern-based approach for extracting fine-grained semantic relations between verbs (*similarity*, *strength*, *antonymy*, *enablement* and *happens-before*). This inventory is different from ours, especially it does not include relations such as entailment and presupposition with complex inferential behavior. For a sample of 100 automatically labeled verb pairs they determined a precision of 65.5%.[59] Results for recall and F_1-score were not reported.

The only common class of semantic relations used by both approaches is *antonymy* or *opposition*. We investigated the verb pairs which are labeled with this class for both systems, comparing to our gold standard test set. Most antonyms are annotated by both systems correctly. Evaluating both systems against our test set yields 71% precision and 35% recall for VerbOcean. With 72% precision and 74% recall our system achieves better recall and overall more balanced results. Examples of verb pairs which could not be found in VerbOcean are *(hide, show)* or *(multiply, divide)*. Some of the antonyms were annotated in VerbOcean with the class *similar*, e.g. *(marry, divorce)* or *(play, work)*. We also find some verb pairs for which VerbOcean performs better than our system, e.g. *(catch, miss)*. With an overall F_1-score of 55% with balanced precision and recall obtained on a more difficult and more balanced data set, our results can be considered competitive.

Inui et al. (2005) perform classification of causal relations for Japanese. They report high precision and recall results (95% precision for *cause, precond* and *means* relations with 80% recall and 90% precision for *effect* with 30% recall). They emphasize that the framework can be applied to other languages, such as English, but no experiments are presented in the paper.

[57] For coreference resolution we employed the Stanford CoreNLP resolver (Lee et al., 2011) – the system that performed best in the 2011 CoNLL Shared Task on coreference resolution.

[58] We check for the presence of the Stanford parser dependency *AMOD* (adjectival modifier) between a verb and a noun (de Marneffe et al., 2006) as an indicator of erroneous annotation.

[59] Only 2 and 8 pairs were evaluated for *enablement* and *antonymy*, respectively.

Pekar (2008) performs acquisition of verb entailment rules. His main focus is on detecting asymmetric relations between verbs and relating their argument positions, without trying to sub-classify the obtained verb pairs into different relation types. His method is based on co-occurring verbs within locally coherent text and measures their asymmetric dependence using an information theoretic approach. A precision of 71% is reported, but no recall and F_1-score.

Acquisition of verb entailment rules is also the aim of Aharon et al. (2010). They acquire inference rules from the FrameNet resource using frame-to-frame relations and induce argument mappings for the related predicates. The obtained rules are tested against ACE events. Performance results are mixed, with modest precision and very low recall: precision: 55.1%, recall: 17.6%, F_1-score: 24.6%.

Berant et al. (2010) explore graph optimization using integer linear programming (ILP) in order to find the best set of entailment rules under a transitivity constraint. The approach is restricted to entailment relations. They obtain balanced precision and recall at 69.6% and 67.3%, respectively. Their work establishes that global methods outperform local methods for learning entailment relations. Berant et al. (2012) offer extensive evaluation and further refinements of this method.

Weisman et al. (2012) use a large set of linguistically motivated features to acquire verb entailment rules. This feature set is designed to extract a wide spectrum of rules, therefore the system achieves a good recall of 71% with a moderate precision of 40% for the recognition of entailment rules. No attempt is made to distinguish the different relation types acquired by the system.

6. Summary and Conclusions

In this contribution we presented a corpus-based approach for discriminative analysis and classification of fine-grained semantic relations between verbs. The set of relations we consider comprise the non-taxonomic inferential relations *entailment*, *presupposition* and *temporal inclusion*, and the taxonomic relations *antonymy* and *troponymy*. We group *temporal inclusion* and *troponymy* given they have similar inferential properties, and excluded *synonymy* as a result of the nature and technicalities of our corpus-based approach. To the best of our knowledge, we are the first to investigate *presupposition* in a corpus-based lexical semantic relation acquisition task.

The focus of this paper was to analyze the underlying properties of the selected relations, the design of features for a corpus-based learning approach, and to discuss possibilities for the annotation of such fine-grained semantic relations. We present experiments for automatic classification of the target relations with evaluation against the gold standard data set we constructed.

In contrast to prior work, we present an in-depth analysis of the relations we aim to sub-classify, including a characterization of their inferential behavior. We determine a small set of differentiating properties relating to negation and temporal sequence properties. These do not only provide differentiating features for classification. They are also essential for appropriate inference in context, which is the ultimate goal of our work.

Inclusion of the presupposition relation is what clearly distinguishes our work from the state of the art in this area, which primarily focuses on the discovery of entailment relations proper. The acquired pairs of presupposition-triggering verbs and their presuppositional relata encode valuable commonsense knowledge about typical verb sequences and preconditions holding between events, such as *play – win*, *read – cite*, *learn – master* or *hire – fire*. These are not broadly covered in verb lexicons such as WordNet and only found with selected scenario frames in FrameNet (Fillmore et al., 2003). Related work by Chambers and Jurafsky (2008, 2009), which aims at acquiring typical event sequences from large corpora, detects frequently occurring verb pairs, yet does not differentiate between fine-grained relation types. Regneri et al. (2010) learn script-like knowledge using sequences of events they gathered from crowd-sourcing. However, script-like knowledge is only applicable to a small set of typical event chains. Their work relies on gathering event sequences for pre-specified situation types. Our approach is more general, as it is able to learn presupposition and other clearly distinguished inferential relations holding between any verb pairs, using a small set of manual annotations. Finally, our work targets the temporal and inferential differences between the various relation types that are crucial for applying the learned relations in context and for drawing valid inferences.

Our analysis shows that the selected relations can be fully discriminated by their inferential and temporal properties. However, this does not mean that automatic or manual labeling of such verb relations is a trivial task. The classification of fine-grained semantic relations between verbs presents a major challenge, due to complicating factors such as verb ambiguity, coreference of arguments and the complexity and subtlety of the inference properties associated with such relations. This was clearly brought out by our initial annotation experiments that followed traditional annotation strategies: type-based annotation forces annotators to consider complex inferential patterns for (pairings of) different verb meanings out of context; token-based annotation is difficult because the contexts are often involved, with shades of meaning that make decisions difficult. Moreover, acquiring sufficient numbers of context-based annotations is expensive, and it is difficult to ensure that all relevant readings are appropriately represented.

We therefore designed a novel annotation setup that addresses the specific problems we identified: (i) controlling for verb readings and ambiguity, (ii) the need for abstraction from specific contexts and (iii) the need to reduce the complexity of the inferential patterns that need to be checked.

The first two problems are addressed by providing verb pairs with prototypical arguments derived from selectional preference classes. From these representations we automatically generate skeleton sentences offered to the annotators. This restricts the interpretation of the verbs and at the same time provides sufficient generalization from particular contexts. The third problem is addressed by designing a question-based annotation scheme. The complex annotation decisions are broken down to basic decision units and are presented in the form of automatically generated skeleton phrases, with placeholders for prototypical arguments. With this novel setup, we obtain reliable inter-annotator agreement and are able to create a gold standard for evaluating fine-grained semantic relation classification. Our novel question-based annotation scheme relieves the annotator from considering several non-trivial decisions in a single annotation step, and thus holds potential for crowd-sourcing the annotation task to non-experts, in order to acquire larger annotated data sets. However, presenting our task to a non-expert annotator did not confirm these expectations. More adaptations are needed to open up this task for crowd-sourcing.

Having successfully addressed the difficulties of manual annotation, we presented a method for corpus-based acquisition of fine-grained semantic relations between verbs, embedded in a discriminative classification task. The classification model is inspired by the temporal and inferential properties we established for the targeted relations, and are enhanced with corpus-based features designed to detect surface contiguity and semantic relatedness of verb co-occurrences.

The classification makes use of type-based distributional features that are generalized from corpus samples. For this reason, the annotation of training and test data sets can rely on type-based annotations that can be quickly acquired – now that the annotation process has been clarified. Our classification model achieves good results with a small training set comprising about 10 verb pairs per relation.

We proposed two classification architectures: flat and hierarchical classification. Hierarchical classification outperforms flat classification by a margin of 4 points in F_1-score, though not significantly. Both classification architectures achieve significant performance (up to 100% improvement) over a best-feature baseline. These results are still open for improvement, but with an overall performance of 55% F_1-score we are able to show that – despite the considerable complexity of the task – both manual and automatic classification are feasible.

The individual results indicate that *presupposition*, *entailment* and *temporal inclusion* are more difficult to classify than *antonymy*; we also found *presupposition* to outperform *entailment*, yielding higher precision. This effect might be due to the more prominent specific negation properties associated with *presupposition*. Closer investigation of the feature impact shows that temporal properties are most effective for the recognition of the inferential relations *presupposition, entailment* and *temporal inclusion*, while relatedness features are strongest for *antonymy* and the *unrelated class*. The negation features are most effective for identifying *presupposition*.

The analysis of the experiment results offers avenues for further enhancements. Coming up with better solutions for sense disambiguation and coreference resolution could help to eliminate major sources of observed errors. Elimination of noise in preprocessing could further improve the results. The hierarchical classification architecture still suffers from error propagation effects that could be reduced through a collective classification approach. Finally, with only 10 seed verb pairs per relation our current model is weakly supervised. Given that we do not require extensive annotations on the token level, adding more verb pairs for training could further improve the results.

In future work we will apply the learned relations to trigger verbs appearing in context to infer implicit information. For the proper usage of the acquired inference rules we need to disambiguate the candidates for trigger verbs. While prior and current work on textual inference focusses on entailment, we consider in particular the presupposition relation, which is ubiquitous in texts and subject to special conditions regarding temporal sequence and negation properties.

Acknowledgements This work represents a completely revised and extended version of a paper presented at the 2011 DGfS-Workshop *"Beyond Semantics: Corpus-based investigations of pragmatic and discourse phenomena"*. We are grateful for comments and suggestions by the workshop participants, as well as valuable feedback by the anonymous reviewers and the editors. Particular thanks go to our annotators: Katarina Boland, Lukas Funk, Jan Pawellek and Carina Silberer.

References

Ben Roni Aharon, Idan Szpektor, and Ido Dagan. Generating entailment rules from FrameNet. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 241–246, Uppsala, Sweden, 2010.

James F. Allen. Maintaining knowledge about temporal intervals. In *Communications of the ACM*, pages 832–843. ACM Press, November 1983.

Marco Baroni and Alessandro Lenci. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK, 2011.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The WaCky Wide Web: a collection of very large linguistically processed web-crawled corpora. In *Journal of Language Resources and Evaluation*, volume 43(3), pages 209–226, 2009.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1220–1229, Uppsala, Sweden, 2010.

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

Steven Bethard and James H. Martin. CU-TMP: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval ’07, pages 129–132, Prague, Czech Republic, 2007.

Johan Bos. Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. *Computational Linguistics*, 29(2):179–210, 2003.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of the ACL/HLT 2008 Conference*, pages 789–797, Columbus, Ohio, 2008.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore, 2009.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, Prague, Czech Republic, 2007.

Timothy Chklovski and Patrick Pantel. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40, Barcelona, Spain, 2004.

David R. Clausen and Christopher D. Manning. Presupposed content and entailments in Natural Language Inference. In *Proceedings of the 2009 Workshop on Applied Textual Inference, ACL-IJCNLP 2009*, pages 70–73, Suntec, Singapore, 2009.

Jacob A. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, pages 37–46, 1960.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(Special Issue 04):i–xvii, 2009.

Christian Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R. L. Petruck. Background to Framenet. *International Journal of Lexicography*, 16(3):235–250, 2003.

John R. Firth. A synopsis of linguistic theory 1930–1955. *Studies in linguistic analysis*, pages 1–32, 1957.

Anette Frank and Sebastian Pádo. Semantics in computational lexicons. In Claudia Maienborn, Klaus Heusinger, and Paul Portner, editors, *Semantics: An International Handbook of Natural Language Meaning*, volume 3 of *HSK Handbooks of Linguistics and Communication Science Series*, pages 2887–2917. Mouton de Gruyter, 2012.

L. T. F. Gamut. *Logic, Language, and Meaning: Introduction to Logic*. University of Chicago Press, Chicago, 1991.

Michael L. Geis and Arnold M. Zwicky. On invited inferences. *Linguistic Inquiry*, 2(4):561–566, 1971.

Bart Geurts and David Beaver. Presuppostion. In Claudia Maienborn, Klaus Heusinger, and Paul Portner, editors, *Semantics: An International Handbook of Natural Language Meaning*, HSK Handbooks of Linguistics and Communication Science Series. Mouton de Gruyter, 2012.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. Automatic discovery of part-whole relations. *Computational Linguistics*, 32:83–135, 2006.

Marti Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING)*, pages 539–545, Nantes, France, 1992.

Verena Henrich and Erhard Hinrichs. Standardizing Wordnets in the ISO Standard LMF: Wordnet-LMF for GermaNet. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 456–464, Beijing, China, 2010.

Takashi Inui, Kentaro Inui, and Yuji Matsumoto. Acquiring causal knowledge from text using the connective marker tame. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(4):435–474, 2005.

Hans Kamp and Uwe Reyle. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht, 1993.

Lauri Karttunen. Simple and phrasal implicatives. In *Proceedings of *SEM: The First Joint Conference on Lexical and Computational Semantics*, pages 124–131, Montréal, Canada, 2012.

Claudia Kunze and Lothar Lemnitzer. Germanet - representation, visualization, application. In *Proceedings of the 3rd International Language Resources and Evaluation Conference (LREC'02)*, pages 1485–1491, Las Palmas, Canary Islands, 2002.

Chong Min Lee. Temporal relation identification with endpoints. In *Proceedings of the NAACL/HLT 2010 Student Research Workshop*, pages 40–45, Los Angeles, California, 2010.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA, 2011.

Stephen C. Levinson. *Pragmatics*. Cambridge: Cambridge University Press, 1983.

Dekang Lin and Patrick Pantel. Discovery of inference rules for Question Answering. *Natural Language Engineering*, 7:343–360, 2001.

John Lyons. *Semantics*. Cambridge University Press, 1977.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *In Proc. Intl Conf. on Language Resources and Evaluation (LREC*, pages 449–454, 2006.

Saif Mohammad and Graeme Hirst. Distributional measures of semantic distance: A survey. arXiv:1203.1858v1, 2012. First published in 2006.

Robert Munro, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily. Crowdsourcing and language studies: the new generation of linguistic data. In *Proceedings of the NAACL/HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 122–130, Los Angeles, California, 2010.

Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia, 2006.

Viktor Pekar. Discovery of event entailment knowledge from text corpora. *Computer Speech & Language*, 22(1):1–16, 2008.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 979–988, Uppsala, Sweden, 2010.

Philip Resnik. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159, 1996.

Tom Richens. Anomalies in the WordNet verb hierarchy. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 729–736, Manchester, UK, 2008.

Rob van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9: 333–377, 1992.

Peter F. Strawson. On referring. *Mind*, 59(235):320–344, 1950.

Galina Tremper. Weakly supervised learning of presupposition relations between verbs. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 97–102, Uppsala, Sweden, 2010.

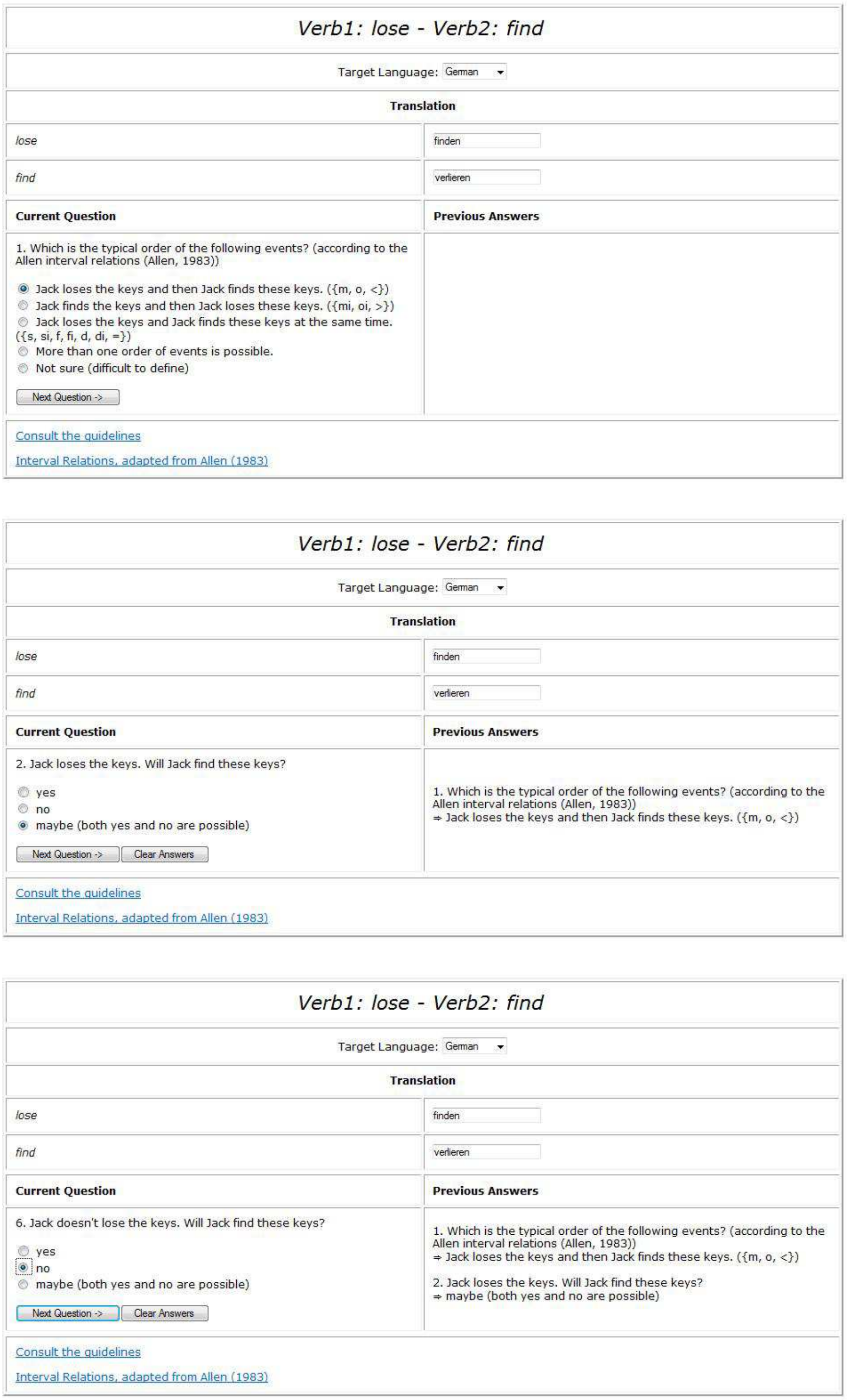
Galina Tremper and Anette Frank. Extending semantic relation classification to presupposition relations between verbs. In *Proceedings of the DGfS Workshop: "Beyond Semantics: Corpus-based investigations of pragmatic and discourse phenomena"*, Göttingen, Germany, 2011.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden, 2010.

Hila Weisman, Jonathan Berant, Idan Szpektor, and Ido Dagan. Learning verb inference rules from linguistically-motivated evidence. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 194–204, Jeju Island, Korea, 2012.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2nd edition, 2005.

Appendix 1. Web-based Annotator Interface



Verb1: lose - Verb2: find

Target Language: German

Translation

lose	finden
find	verlieren

Current Question

1. Which is the typical order of the following events? (according to the Allen interval relations (Allen, 1983))

● Jack loses the keys and then Jack finds these keys. ({m, o, <})
○ Jack finds the keys and then Jack loses these keys. ({mi, oi, >})
○ Jack loses the keys and Jack finds these keys at the same time. ({s, si, f, fi, d, di, =})
○ More than one order of events is possible.
○ Not sure (difficult to define)

Previous Answers

Consult the guidelines

Interval Relations, adapted from Allen (1983)

Verb1: lose - Verb2: find

Target Language: German

Translation

lose	finden
find	verlieren

Current Question

2. Jack loses the keys. Will Jack find these keys?

○ yes
○ no
● maybe (both yes and no are possible)

Previous Answers

1. Which is the typical order of the following events? (according to the Allen interval relations (Allen, 1983))
⇒ Jack loses the keys and then Jack finds these keys. ({m, o, <})

Consult the guidelines

Interval Relations, adapted from Allen (1983)

Verb1: lose - Verb2: find

Target Language: German

Translation

lose	finden
find	verlieren

Current Question

6. Jack doesn't lose the keys. Will Jack find these keys?

○ yes
● no
○ maybe (both yes and no are possible)

Previous Answers

1. Which is the typical order of the following events? (according to the Allen interval relations (Allen, 1983))
⇒ Jack loses the keys and then Jack finds these keys. ({m, o, <})

2. Jack loses the keys. Will Jack find these keys?
⇒ maybe (both yes and no are possible)

Consult the guidelines

Interval Relations, adapted from Allen (1983)

Figure 5: Question-based Annotation for verb pair *lose* – *find*.

Appendix 2. Mapping of Allen's Relations to Coarse Temporal Relation Classes

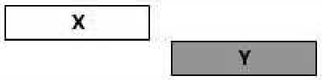
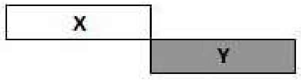
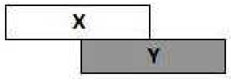
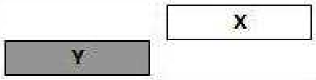
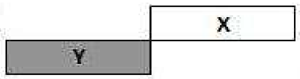
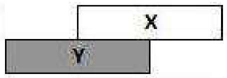
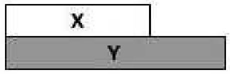
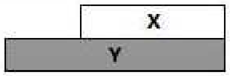
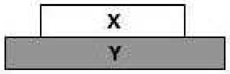
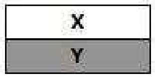
Temporal Relation Class	Allen's Relation	Graphical Representation
before(X,Y)	X < Y (strict precedence)	
	X m Y (X *meets* Y)	
	X o Y (X *overlaps* Y)	
after(X,Y)	X > Y (strict succession)	
	X mi Y (inverse of meets)	
	X oi Y (inverse of overlaps)	
during(X,Y)	X s Y (X *starts* Y)	
	X f Y (X *finishes* Y)	
	X d Y (X *during* Y)	
	X = Y (X *equals* Y)	

Table 12: Mapping of Allen's Relations