博士論文

Interlingual Semantic Analysis of Text: Alternative methods to full corpus annotation

（テキストの中間言語方式意味的解析：全コーパスアノテーションの代替法）

Andre Kenji Horie
ホリエ　アンドレ　ケンジ

THE UNIVERSITY OF TOKYO

Interlingual Semantic Analysis of Text: Alternative methods to full corpus annotation

by

André Kenji Horie

Thesis submitted in partial fulfillment for the degree of
Doctor of Philosophy

in the
Graduate School of Information Science and Technology
Department of Creative Informatics

August 2014

Abstract

In Natural Language Processing, a trend towards shallow linguistic statistical approaches has been observed recently. While these approaches have low implementation costs and present reasonably satisfactory outputs, there is a trade-off on the quality and naturality of the output. Deep semantic approaches, on the other hand, enable meaning to be better conveyed, despite requiring manual annotation of large corpora to be used by supervised machine learners.

The high cost of manual annotation represents one of the main reasons hindering a wider adoption of deep semantic approaches. This research thus aims to decrease such costs by: considering semantic domains separately, following the Principle of Compositionality; analyzing language-independent semantic elements that constitute contextualized events for each of the selected domains of contextual relations, modality and tense; and proposing alternative methods which aim to decrease the number of annotated instances, simplify annotation and/or decrease requirements on annotators' level of specialization.

For the domain of contextual relations, bootstrapped set expansion is proposed. It is a semi-supervised semi-automatic process which extracts new instances from an unlabeled corpus such as the web to be manually annotated. This approach addresses the problems of class imbalance and incomplete feature spaces, creating feature-rich datasets from initial small seeds and enabling better allocation of annotation resources.

For the domain of modality, cue expression disambiguation and selection are decoupled. This enables optimization of the latter, which in turn renders the non-interlingual cue annotation unnecessary. It is empirically shown that selection outperforms existing systems in the general case even when using less resource-intensive disambiguation settings such as lemmatization only.

Finally, for the domain of tense, this research first and foremost presents the novel task of semantic analysis of tense using sequential classification. The cost of annotation of such approach is then decreased by introducing more intuitive descriptors and by automatically inferring tense, which is only possible because of the proposed theory of tense. Unlike other works, which assume extraction of semantic structures as trivial, this theory investigates how tense is perceived and composed from surrounding temporal entities (verbs, adverbials and context), which is evaluated through a proof of concept.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisors during the doctorate program. This dissertation was only possible because of the invaluable guidance and support from Prof. Kumiko Tanaka-Ishii, Prof. Masatoshi Ishikawa and Prof. Mitsuru Ishizuka.

I would also like to extend my thanks to: professors and staff from the Department of Creative Informatics and the Graduate School of Information Science and Technology for the assistance throughout these years; the Government of Japan represented by its Ministry of Education, Culture, Sports and Technology, for the scholarship which provided me the necessary financial support for my stay in Japan; and to members, former members and staff from the Tanaka, Ishikawa and Ishizuka Laboratories, who have helped me in many different ways.

Last but not the least, I sincerely thank my wife Elis, my mother Satie, my brothers Hideki and Koji, my father Yuji, and all of my relatives and friends for their support and encouragement.

Dedicated to my wife Elis
and in loving memory of my mother Satie

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Interlingual Semantic Computing

Semantics, as the object of study of Linguistics, regards explaining the intrinsic meaning of words and phrases by giving model-theoretical interpretations to fragments of natural language (NL) with the help of some intermediate level of logical representation. *Semantic Computing* (also referred to as *Computational Semantics*) regards computationally building such representations, denominated *semantic structures* or *semantic constructions*, and reasoning with the results [Blackburn & Bos, 2003]. *Interlingual Semantic Computing* is the Natural Language Processing (NLP) task which extracts language-independent semantic structures from texts. These structures are named *interlingua*, term that refers to the intermediate artificial language which acts as bridge between texts in different languages.

1.1.1 The Interlingual Paradigm

Language can be divided into layers according to the depth of information density [Vauquois, 1968], as illustrated in figure 1.1.

Shallower linguistic approaches are concerned with natural language symbols and their groupings. Strings of characters are separated into sentences and words, and the surface forms of such words may be encoded into bags-of-words or n-grams to be used by statistical methods.

Intermediate approaches are concerned with morphology, which explains the structure of individual words, and with syntax, which explains formal language structure and phrasal hierarchies.

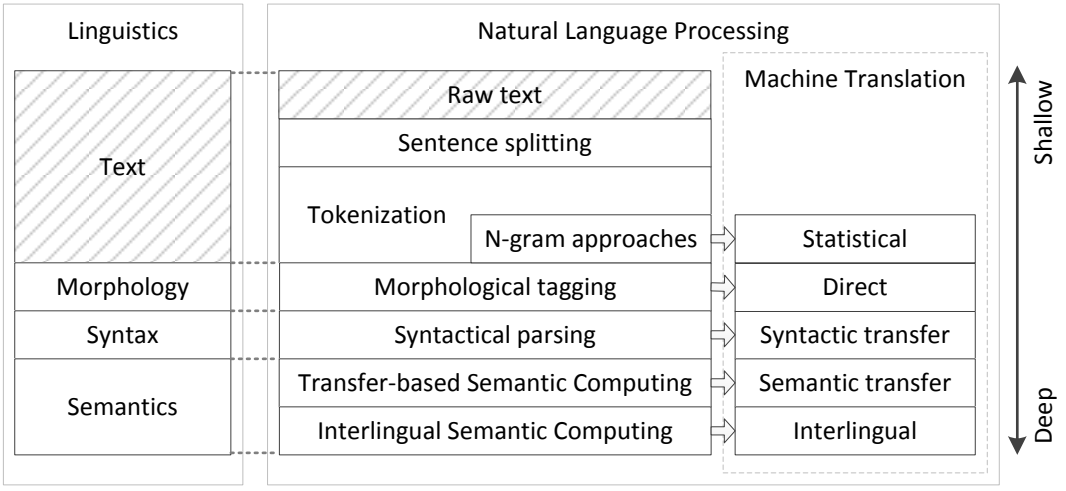

FIGURE 1.1: Layers of information in Linguistics with corresponding NLP tasks

Finally, deeper linguistic approaches are concerned with data structures representing meaning and speaker intent encoded within NLs. This representation may be language-dependent for transfer-based semantic approaches, or language-independent for interlingual approaches.

An example of application of these NLP paradigms is given for machine translation [Dorr et al., 1999]. For shallow linguistics approaches, *Statistical Machine Translation* (SMT) [Koehn, 2010] uses probabilities on the occurrence of n-grams in parallel corpora, and *direct translation* translates a text word-by-word or phrase-by-phrase according to a dictionary and to a set of pre-defined rules. For intermediate approaches, *syntactic transfer* performs syntactic reordering of words on top of the direct translation. Finally, for deep linguistic approaches, *semantic transfer* performs translation based on meaning for a given language pair, whereas *interlingual translation* uses semantic structures which are independent of source and target languages.

1.1.2 Interlingual Analysis

In Interlingual Semantic Computing, the interface with NLs is obtained through two tasks: the *analysis task*, which converts texts from NLs to an interlingua, and the *generation task*, which converts texts from an interlingua to NLs. Interlingua is specified by a *schema* based on linguistic theories of semantics on some fragment of NLs, herein denominated *semantic domain*. This is illustrated in figure 1.2.

For the interlingual analysis task, supervised machine learning techniques have been widely and successfully employed. In supervised learning, a *classifier* learns a *generic model* from *training data* structured according to the pre-defined interlingua schema, in a step called *training*; and applies this model to unseen raw data in order to structurize it,

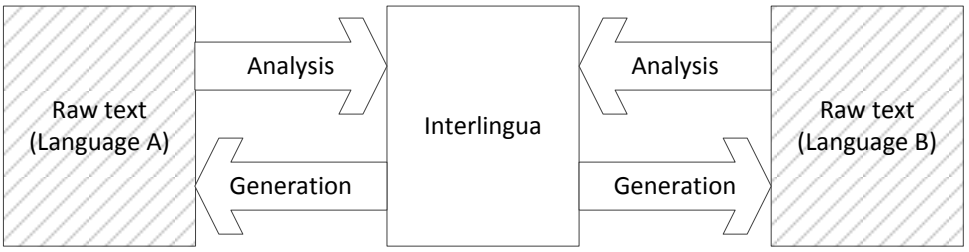

FIGURE 1.2: Tasks in Interlingual Semantic Computing

in a step called *classification*. Training data for supervised learning needs to be obtained by means of human effort, in a process called *manual annotation*. This is illustrated in figure 1.3.

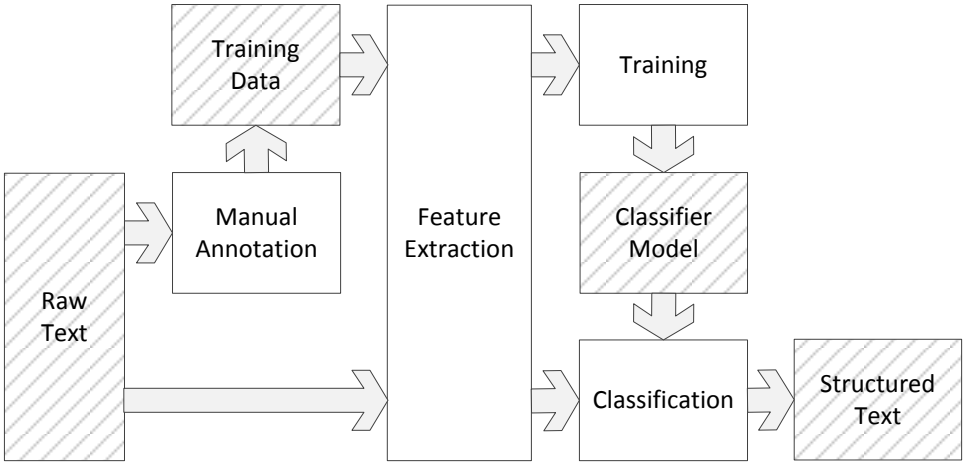

FIGURE 1.3: Overview of semantic analysis using supervised machine learning

Each computational unit from both training and testing data is called an *instance*. The exact definition of instance depends on the granularity level of the knowledge representation schema used.

Each instance is described by information called *linguistic features*. Some normally used features are briefly described:

- *Part-of-speech (POS) tags*: Morphological features that indicate the class of a word. More information is given in appendix A.
  Ex.: The word "*Peter*" is a proper noun in the singular, receiving the tag NNP
- *Phrase structure trees* [Chomsky, 1957] and *dependency trees* [Tapanainen & Järvinen, 1997]: Syntactic features that indicate the grammatical behavior words within the syntactic structure of text. Because they are both syntactic constructs, there is a correspondence between the two types of trees. More information is given in appendix B.

Ex.: The dependency between words "*I*" and "*go*" in the sentence "*I will go to Osaka*" is nominal subject (nsubj)

- *Named entity (NE) tags*: Lexical features that classify proper nouns into people, places and institutions.
  Ex.: The word "*Peter*" is a person's noun, and thus would receive the tag `PERSON`, whereas "*Japan*" would receive the tag `PLACE` and "*University of Tokyo*" would receive the tag `INSTITUTION`

- *Word senses*: Lexical features that indicate possible meanings of an uncontextualized word according to a dictionary or other body of knowledge. The NLP task of identifying the correct sense of a contextualized word is denominated Word Sense Disambiguation (WSD).
  Ex.: Some senses for the noun "*bank*" are "*financial institution*", "*land alongside a river*", "*container for keeping money at home*", etc.

- *Ontological classifications*: Ontological features that regard the hierarchy of words according to a dictionary or other body of knowledge.
  Ex.: "*Bank*" is the same as "*banking company*" and is a type of "*financial institution*".

The way these features are encoded depend largely on the application and on the type of input accepted by the machine learner. For example, consider WordNet [Fellbaum, 1998], a linguistic resource which provides word sense, hypernymy, hyponymy, and other lexical and ontological information. A WordNet entry is illustrated in figure 1.4, in which the sense "*financial institution*" for the word "*bank*" (identified as sense `bank#1`) is given, with each row representing a layer in the hypernymy hierarchy described by the entry ID, a general ontological classification and synonym senses represented by the word concatenated with the sense ID. Such an entry can be encoded by its ID, by the path to another entry within the hierarchy tree, among others.

{08305900} <noun.group> bank#1, depository financial institution#1, banking concern#1, ...
  {07945061} <noun.group> financial institution#1, financial organization#1, ...
    {07943952} <noun.group> institution#1, establishment#2
      {07899136} <noun.group> organization#1, organisation#3
        {07842951} <noun.group> social group#1
          {00029714} <noun.Tops> group#1, grouping#1
            {00002236} <noun.Tops> abstraction#6
              {00002119} <noun.Tops> abstract entity#1
                {00001740} <noun.Tops> entity#1

FIGURE 1.4: Example of WordNet ontology for one sense of the noun "*bank*"

1.1.3 Problems with Interlingual Analysis

Despite early adoption in machine translation, interligual approaches have been replaced in favor of shallow linguistic approaches. While interlingua provides semantic richness, which enables NLP services with more quality, its analysis step is costly. In addition, it has been questioned if a pure interlingual system is possible at all. These two problems are detailed and further discussed in the following sections.

1.2 Addressing the Open-Domain Problem

A pure interlingual system requires that meaning in its entirety is abstracted into a data structure by the analysis step, removing all morphosyntactic and language-dependent semantic components. The interlingua in this case is denominated *open-domain interligua*. However, it is widely regarded that defining a schema for an open-domain interligua is not feasible, to say the least, because it requires complete knowledge on the mechanisms of all semantic phenomena for every existing language. In addition, these phenomena are structured in completely different ways, making the task of proposing a generic theory for semantics not trivial. As a very simplistic example, some parts of semantics may be modeled as a multi-level directed graph, whereas others as positioning elements in the time axis.

1.2.1 Compositionality of Meaning

Instead of attempting to unify Semantics, another more successful approach is to separate its components. The Principle of Compositionality [Frege, 1892] states, in simple terms, that the meaning of every expression e in a language L is determined by the structure of e and the meanings of the constituents of e in L. By studying these constituents separately, it is possible to isolate parts of meaning that are important for specific applications. For example, *minimally descriptive interlinguas* have been used for concept description in open-domain texts (section 2.1.3), representing knowledge under a framework with limited semantic expressiveness.

Compositionality thus justifies the study of individual semantic domains. Some well-established ones are then briefly presented in the following section. It should be noted that decomposing semantics does not enable pure interlingual systems to be built, but it provides a source of deeper linguistic information, as it is discussed later in section 1.4.

1.2.2 Semantic Domains

First and foremost, semantic domains may be classified according to two categorizations: based on contextualization and on granularity.

Contextualization refers to whether meaning is considered within or without a textual context. A *lexical semantic* domain refers to the meaning of words or expressions outside of context (the vocabulary of a language), being static or slowly evolving by nature. A *context-sensitive lexical semantic* domain refers to the meaning of words or expressions according to a given context. Finally, a *propositional semantic* domain refers to meaning of clauses or sentences within the text.

Granularity refers to the scale or level of detail of the element taken into consideration. First, consider the following definitions of entity, relation and event:

- *Entity*: Word or group of words that expresses any given concept or set of concepts. An entity can be morphosyntactically represented by as little as one single word, but it can also be represented by a group of words or even a group of connected entities.
  Ex.: The words "*bank*" and "*large bank*" are entities
- *Relation*: An irreflexive connection between any two entities, named *head entity* and *tail entity*, described in terms of common attributes which form the basis of a *relation class*.
  Ex.: The words "*Tokyo*" and "*Japan*" share a relation of class Capital-Country
- *Event*: A super-entity containing at least one sub-entity which represents an action or state. Actions and states are enclosed within a clause, and are usually morphosyntactically indicated by a verbal complex, although other constructions are possible.
  Ex.: "*I will go to Osaka*" is an event

Possible granularities considered are: the *entity level*, when the semantics of a single non-event entity is concerned; the *inter-entity level*, when the semantics of the relation between two entities which do not compose an event is concerned; the *intra-event level*, when the semantics of the relation between two entities which compose an event is concerned; and the *inter-event level*, when the semantics of the relation between two events is concerned.

A list of semantic domains according to the previous categorization is then presented in table 1.1, with brief descriptions of each domain given in the following sections.

TABLE 1.1: List of semantic domains

	Lexical Semantics	Context-Sensitive Lexical Semantics	Propositional Semantics
Entity level	• Polysemy • Homonymy		
Inter-entity level	• Uncontextualized lexical relations	• Contextualized lexical relations • Quantification	
Intra-event level		• Semantic roles • Tense • Aspect	• Modality • Negation • Focusing
Inter-event level			• Discourse relations • Paraphrasing • Contradiction

This list is by no means exhaustive. It is observed that: lexical semantics does not relate to the event level, since events are necessarily contextual; context-sensitive lexical semantics does not relate to the entity and the inter-event levels, since it requires a relation for the contextual reference, but may not refer to events directly because of the lexical nature; and propositional semantics does not relate to entity and inter-entity levels, since it is based on events.

1.2.2.1 Contextualized Semantic Relations

Contextualized semantic relation is a generic term for all relations among entities and/or events that occur within a context. It is comprised of:

- *Contextualized lexical relations*: Lexical relations presented within a context.
  Ex.: The synonymy relation in sentence "*'attractive' is a synonym of 'beautiful'*"
- *Semantic roles* (also known as *thematic roles*): Relations among the constituents of an event.
  Ex.: The word "*I*" starts the action of "*go*" in the sentence "*I will go to Osaka*"
- *Discourse relations*: Relations which connect events throughout the text. Some notable discourse relation classes are *paraphrasing*, which indicates the restatement of the meaning of a text or passage using other words, and *contradiction*, which indicates statements that are opposed to one another.
  Ex.: The reason relation between the two clauses of the sentence "*I am tired because I couldn't sleep tonight*"

1.2.2.2 Tense, Aspect and Modality

Tense, aspect and modality are highly correlated semantic domains which regard properties of an event, being normally marked within the verbal complex of a clause. *Tense* refers to the time of the event, *aspect* to the nature of the event in terms of temporal constituency, and *modality* to the status of the proposition that describes the event [Binnick, 1991; Palmer, 2001].

Some examples are the future tense in sentence 1.1a, the habitual aspect in sentence 1.1b and obligation in the modalized sentence 1.1c. Other examples of tenses are: past, present, past perfect, etc.; examples of aspects are: stative, progressive/continuous, gnomic (absolute truths, ex.: "*The sun rises in the east*"), etc.; and examples of modality are: uncertainty, volition, etc.

(1.1) a. I will go. — Tense: future

b. I always go. — Aspect: habituality

c. I must go. — Modality: obligation

1.2.2.3 Other Domains

Two examples of lexical semantic domains at the entity level are polysemy and homonymy. *Polysemy* represents the multiple possible related meanings a word or expression might possess, ex.: the word "*man*" may refer to the human species (opposite of "*animal*"), to the male individual of the human species (opposite of "*woman*") and to the adult male individual of the human species (opposite of "*boy*"). On the other hand, *homonymy* represents the multiple possible unrelated meanings a word or expression might possess, ex.: the word "*skate*" may refer to gliding on ice or to a type of fish. The set of words and concepts of a given NL with respective polysemes and homonyms is named a *lexicon*.

Uncontextualized lexical relations represent the connection between two semantic entities in terms of common attributes. Some notable relation classes are *synonymy* (Is-Same-As relation), *antonymy* (Is-Opposite-Of relation), *hypernymy* (Is-A-Type-Of relation) and *hyponymy* (Is-A-Generalization-Of relation). For example, "*happy*" is a synonym of "*joyful*" and an antonym of "*sad*", and "*cat*" is a hypernym of "*feline*" and a hyponym of "*domestic cat*". Uncontextualized relations are also used for building bodies of knowledge such as *ontologies*, which provide a hierarchy of concepts for a language.

Quantification represents the semantics of specification of a set, indicating the number of objects from a certain set which are valid. For example, given the set of dogs D and

the set of domestic things T, the sentence "*Most dogs are domestic*" can be interpreted as the intersection between D and T being greater than the remainder of D.

Negation is an operator which functions along with quantifiers and modals, inverting the truth value of a proposition [Trask, 1993]. For example, "*I didn't see anything*" is the negation for "*I saw something*". It is observed that negation interacts with other operators in complex ways that vary from language to language. For example, the Spanish translation for the previous example is "*No he visto nada*" (lit. "*I didn't see nothing*"), in which the negation requires other elements to be marked as negative.

Some other worth-noticing domains regard to Pragmatics, the subfield of Linguistics and Semiotics which studies how the context of the utterance and any pre-existing knowledge contribute to meaning. *Assertion* is a statement based on evidence, whereas *pressuposition* regards an implicit assumption. *Politeness* is the linguistic mechanism through which tact, generosity, approbation, modesty, agreement and sympathy are conveyed in conversations [Leech, 1983]. Finally, *entailment* is when the truth value of one sentence implies the truth value of another. For example, "*Mary loves flowers*" implies "*Mary loves roses*".

1.3 Addressing the Cost Problem

As presented in section 1.1.2, supervised machine learning has been employed for Semantic Computing. Given a manually annotated training dataset, a classifier learns a generic model from such data. As a result, the performance of classifiers is largely dependent on the quality of dataset provided. First and foremost, some desired properties of manually annotated datasets are described:

- *Representativeness*: Annotated data must be a fair representative of a text genre or knowledge domain. If training data does not represent the nature of testing data, the model obtained by the machine learner will not be a proper descriptor
- *Completeness*: Annotated data must thoroughly cover as many feature combinations as possible, decreasing the number of unknown features combinations in unseen instances
- *Correctness*: Data is expected to be correctly annotated in order to avoid propagation of error to automatically annotated data
- *Consistency*: Manual annotation should follow guidelines consistently throughout the corpora for all annotators. All of the following forms of consistency are required:

- Intra-annotator consistency: When instances annotated by one individual are consistent among themselves
- Inter-annotator consistency: When instances annotated by different individuals are consistent among themselves
- Cross-language consistency: A special case of inter-annotator consistency for interlingual approaches, in which instances annotated in different languages are consistent among themselves

Representativeness is normally obtained by means of the choice of training dataset, choosing a corpus whose instances are similar to the ones to be analyzed by the application; completeness by increasing the size of the dataset; correctness by providing proper instruction to the manual annotators; and consistency by effort on decreasing annotation ambiguity, peer-checking, among others.

1.3.1 Annotation Cost

In order to obtain the previously described properties, several factors need to be taken into consideration, such as the choice of the corpus to be annotated, its size, the ambiguity and intuitivity of the annotation schema, etc., all of which impact the total cost of annotation.

The annotation cost can then be estimated as follows. Let $\mathcal{T}$ be an annotation task and L be a set of languages. The cost $\mathcal{C}$ for given $\mathcal{T}$ and L is given by the equation

$$\mathcal{C}(\mathcal{T}, L) = \sum_{L_i}^{L} T_1(\mathcal{T}, L_i) \cdot N(\mathcal{T}, L_i) \cdot A(\mathcal{T}, L_i) \tag{1.1}$$

where T_1 is the average time to manually annotate one instance for task $\mathcal{T}$ for a given language L_i, N is the number of instances to be annotated, and A is the cost of an annotator per unit of time.

It is noted that this equation does not take into consideration aspects such as cost of consistency checking. While it would not be difficult to include such costs, this work has opted to focus on the main annotation step only.

1.3.2 Annotation Complexity

In order to estimate the total cost of a manual annotation task, since both the number of instances to be annotated N and the cost of annotator per unit of time A are known

beforehand, the average time to manually annotate one instance T_1 becomes the most difficult variable to be determined. It can nevertheless be estimated by averaging the time different annotators take to perform a subset of the annotation task, dividing by the size of the subset.

When comparing two annotation methods, however, using this real cost estimate might not be the best alternative. Some reasons are given:

1. The time to annotate one instance is highly dependent on the individual annotator
2. Annotators get used to the task and problem. As a result, the second method to be compared is expected to be performed faster than the first method
3. Method comparison is normally carried during the prototypying step. As a result, real cost assessment may be costly and slow for this purpose, requiring the definition of a corpus along with formal annotation guidelines

Annotation complexity is then defined by this work as an estimate of the time taken to extract the semantic structure from text in terms of number of annotation decisions given the input size N, for a large N, similarly to computational complexity. Aside from the familiarity of computer scientists with the Big O notation, such approach is proposed because it provides a straightforward qualitative comparison of annotation methods, disregarding possible sources of estimation errors. Using an analogy with computational time complexity, methods are compared by number of instructions executed, not taking into consideration processing power, caching, etc. Some examples of annotation complexity are given in table 1.2.

TABLE 1.2: Examples of annotation complexity for different types of annotation

Type of Annotation	Annotation Complexity (for all N instances)
Binary classification	$\mathcal{O}(2N)$
Multiclass classification (C classes)	$\mathcal{O}(CN)$
Tree structure identification	$\mathcal{O}(NlogN)$
Co-occurrence identification	$\mathcal{O}(N^2)$
Selection (combinatorial decision problem)	$\mathcal{O}(2^N)$

1.3.3 Strategies for Cost Reduction

The total cost of annotation $\mathcal{C}$ is directly proportional to: the annotation time T_1, which depends on factors such as average speed of individual annotators and structural complexity of the annotation schema; the size of the corpus N; and the cost of annotators

A, which depends on localization, level of specialization required, supply and demand of annotators, etc. Consequently, cost reduction of manual annotation is then possible through three different strategies:

(S1) Decreasing T_1 by decreasing annotation complexity or improving annotation tools

(S2) Decreasing N from *full corpus annotation* to a smaller dataset

(S3) Decreasing A by simplifying the way which knowledge is described so that less specialized annotators are able to perform the task

Under the current research trend, generic methods have been favored in NLP due to the wider applicability. The focus has thus been on families of methods under strategy (S2), which allows generic frameworks to be applied to different problems. *Sampling* [Dagan & Engelson, 1995; Engelson & Dagan, 1996] has been used in order to select a subset of instances from the corpus. *Active learning* [Settles, 2009] is a semi-supervised semi-automatic learning paradigm which states that a machine learning algorithm can achieve greater accuracy with fewer training instances if it properly chooses data from which it learns, querying an *oracle* (human annotator) to label instances whose classification would otherwise be difficult. Finally, *co-training* [Blum & Mitchell, 1998] is a semi-supervised automatic approach that requires two views of the linguistic phenomenon, each of which using its own set of features that provides different but complementary information about each instance.

Domain-specific approaches, most of which regard strategies (S1) and (S3), have not been widely attempted. Strategy (S1) concerns a more computational aspect of the problem, decreasing the complexity of annotation or improving annotation tools. It is reported to have been used in fields such as image and ontology annotation [Erdmann et al., 2000; Wenyin et al., 2001], among others. Strategy (S3) concerns a more theoretical linguistic aspect, since the knowledge representation needs to be redefined. The simplification of knowledge representation descriptors has not been an object of research in Linguistics, since novelty for semantic theories is regarded as better descriptive models.

1.4 Motivation and Objectives

1.4.1 Towards Enriching Shallow Linguistics with Deep Linguistics

The ultimate objective of this research is the extraction of semantic structures to be used to enrich statistical models. This application of Interlingual Semantic Computing

is more realistic than a pure interlingual approach, since semantic domains can be used separately, and there is thus no need for an open-domain interlingua. Some works illustrate how shallow linguistics can be enriched by interlinguas, such as the work by Avramidis & Koehn [2008], who showed that factored models for source information improved translation accuracy from morphologically poor to rich languages, and by Wu & Fung [2009], who showed that semantic roles may improve SMT.

A machine translation example of how a deep linguistic approach may improve the quality of service of a shallow approach is then given. Needless to mention, interlingua may improve not only translation, but also other applications such as semantic search, knowledge retrieval, semantic question-answering and summarization. Consider the Japanese sentence 1.2a, whose translation in English is given by 1.2b:

(1.2) a. Kono taikai ni wa kengai no hito made deru rashii.

b. It seems even people from outside the prefecture participate in this tournament.

Using a statistical machine translator, the following sentence 1.3 is obtained when translating sentence 1.2a to English:

(1.3) People outside the prefecture to be available for this tournament.

The best translation in SMT is given by $\hat{t} = \arg\max_t p(s|t)p(t)$, where $p(s|t)$ is the *translation model* and $p(t)$ is the *language model*. In other words, given a sentence in the source language s, the translator finds the best sentence in the target sentence $\hat{t}$ by finding the sentence t with the highest probability. The probability of each target candidate t is the multiplication of: (1) the probability that the set of translated words is indeed the translation of the set of original words regardless of word order (translation model); and (2) the probability of that a given word ordering exists in the target language (language model).

Arguably, it is then possible to enrich such translation by adding semantic information from different sources. One alternative, for example, would be to modify values of $p(s|t)$ and $p(t)$ according to extracted semantic structures. This process of enrichment is illustrated hereon.

First, the incorrect verb choice in the example happens because "*to be available*" is more common in the training parallel corpus than "*participate*" for the Japanese verb "*deru*". The reference to "*taikai*" (tournament) should fix the probabilities by penalizing the

former, but this does not happen due to "*taikai*" and "*deru*" being placed outside the 5-gram window of the language model; in other words, the reference between the two words is lost. The translation in this case can be enriched by restoring this reference in the form of labeling a semantic role, for which "*taikai*" is the target of "*deru*". This results in the following sentence:

(1.4) People outside the prefecture *to participate* in this tournament.

Next, the incorrect verb conjugation choice happens because "*deru*" indicates infinitive, simple present and simple future tenses. By analyzing the context, it is possible to determine that the correct tense is present, and the correct aspect is habituative. As a result, the sentence below would be obtained:

(1.5) People outside the prefecture *participate* in this tournament.

Hearsay evidence is also lost in the translation model. By enriching the process with evidential modality, the original statistical translation can be corrected to the sentence below:

(1.6) *It seems* people outside the prefecture participate in this tournament.

Finally, focusing is not modeled by the translation model. The "*made*" in the original Japanese sentence in a focus indicator, which can be used to generate emphasization of the entity "*people outside the prefecture*":

(1.7) It seems *even* people outside the prefecture participate in this tournament.

1.4.2 This Research

This work assumes the compositional nature of semantics, and proposes cost reduction of the analysis task for some selected domains, namely contextual semantic relations, tense and modality, ultimately aiming a higher quality of service by enriching shallow linguistic methods.

The choice of these three domains is to focus on intra-event interlingual semantics, which enables the description of attributes of the event such as related entities, temporality and speaker attitude. As a result, the following domains are not investigated within

this work: transfer-based semantic domains, such as context-insensitive lexical semantic domains, quantification, negation, focusing, among others; and domains with pragmatic component, such as assertion and politeness.

Given that the prohibitive cost of manual annotation inhibits a wider adoption of semantic applications, this research proposes proofs-of-concept which aim to decrease such high cost of manual annotation. It is outside of the scope, however, the integration of shallow methods and interlinguas from different domains. These proofs-of-concept are specific to and use linguistic properties of each domain, as follows.

For contextual semantic relations, bootstrapped set expansion is proposed. Under this approach, an initial small training dataset is increased by iteratively searching similar relations from an unannotated corpus. This regards strategies (S1) and (S2) for decreasing annotation costs.

For modality, cue selection optimization is proposed. It greatly simplifies annotation by using correlation between cue expressions and sentence-level modalization, regarding strategies (S1) and (S3).

Finally, for tense, first and foremost an analysis task using sequential classification is proposed, as there were no previous works that extracted tense semantics to the best of the author's knowledge. A linguistic theory which supports tense inference is then presented, and automatic inference which enables decrease in annotation cost is performed. This regards strategies (S2) and (S3).

This thesis is structured as stated in the outline in table 1.3. Each following chapter addresses one specific domain: chapter 2 addresses the domain of contextual semantic relations, chapter 3 addresses modality, and chapter 4 addresses tense.

TABLE 1.3: Knowledge representation and analysis tasks for selected domains

Domain	Knowledge Representation	Analysis	Alternative Analysis
Contextual Semantic Relations	PropBank, FrameNet, CDL, etc. (sec. 2.1)	CoNLL-2005 Shared Task, etc. (sec. 2.2)	*Bootstrapped set expansion* (sec. 2.3)
Modality	Cue expression and modal value (sec. 3.1)	CoNLL-2010 Shared Task, etc. (sec. 3.2)	*Heuristic cue selection optimization* (sec. 3.3)
Tense	Reichenbach's framework and extensions (sec. 4.1) *Composite temporal structure* (sec. 4.3)	*Sequential classification* (sec. 4.2)	*Semantic composition* (sec. 4.3)

Current state of research for contextual semantic relations, modality and tense. Proposed work is indicated in *italic*.

For each domain, knowledge representation is described in sections 2.1, 3.1 and 4.1, giving background on how semantics is explained for the particular domain under a linguistic theoretic point of view.

State-of-the-art semantic analysis is then presented in sections 2.2, 3.2 and 4.2, giving details on data annotation efforts, corpora, tasks and systems.

Methods for decrease in annotation cost is proposed in sections 2.3, 3.3 and 4.3. The method is compared in terms of annotation cost and complexity to full corpus annotation, existing domain-specific methods and generic approaches applied to the given domain, whenever applicable.

Finally, discussions and conclusions for each domain are given in sections 2.4, 3.4 and 4.4, and for the entire work in chapter 5.

Chapter 2

Contextual Semantic Relations

2.1 Knowledge Representation

Contextual semantic relations, which are referred to as *semantic relations* henceforth for simplicity purposes, are classified into three types: inter-entity relations, when the relation does not compose an event; intra-event relations, when the relation occurs within a single event; and inter-event relations, when the relation connects two different events.

Inter-entity relations are similar to uncontextualized lexical relations, except that the former are explicitly stated in the text. For example, the synonymy relation between '*attractive*" and "*beautiful*" can be stated explictly as in "*'attractive' is a synonym of 'beautiful'*" or implicitly in a thesaurus.

Intra-event relations are the target of Semantic Role Labeling (SRL) [Carreras & Màrquez, 2004, 2005]. This task associates an action or state with its predicates, classifying them into specific roles which indicate the "who", "what", "when", "where", "how", "by whom", etc. of an event. For example, the entity "*I*" is the starter of the action "*go*" in the sentence "*I will go to Osaka*". These relations are roughly equivalent to syntagmatic relations [de Saussure, 1916; Asher, 1994].

Inter-event relations are the target of discourse analysis and parsing [Marcu, 2000]. This task associates two events by a *discourse relation* (also known as *rhetorical relation*), which indicates how two segments of discourse are connected to each other either at the logical level (ex.: equivalence, condition, contradiction and cause-and-effect) or at the textual level (ex.: cohesion and coherence). For example, in "*U.S. Trust (...) has faced intensifying competition from other firms (...). As a result, U.S. Trust's earnings have been hurt*", the second sentence indicates a consequence of the event stated in the first

sentence. These relations occur beyond the sentence boundary and are equivalent to semantic relations at higher levels of text [Khoo & Na, 2007].

The tasks of SRL and discourse parsing, as well as CDL, an effort which defines relations on all of these levels of granularities, are described in the following sections.

2.1.1 Semantic Role Labeling

Initial efforts on SRL are attributed to Fillmore [1967]. The proposed *case theory* analyzes the valence (number of subjects, objects, etc.) of a verb and its surrounding context, attributing roles *agent*, *object*, *benefactor*, *location* and *instrument* to specific verbs. For example, the verb "*give*" requires an agent (`A`), an object (`O`) and a benefactor (`B`), as shown in the following sentence:

(2.1) Jones (`A`) gave money (`O`) to the school (`B`).

Each verb sense has a set of roles which forms its *case frame*. These roles are subject to some constraints, such as obligatory and optional cases. Example 2.2a illustrates an ungrammatical sentence generated by the omission of the benefactor, an obligatory case for the verb "*give*", whereas example 2.2b illustrates a grammatical sentence for frame "*give away*", which does not require a benefactor.

(2.2) a. * Mary (`A`) gave those apples (`O`).

b. Mary (`A`) gave those apples (`O`) away.

A fundamental hypothesis is that semantic roles have a correlate construction at the syntactic level. For example, the agent role is usually indicated by the subject in active voice or the object in passive voice, though other constructs are possible.

The concepts introduced by case theory are used throughout works on SRL. Three notable efforts, namely PropBank, FrameNet and VerbNet, are then presented, with their merits and demerits discussed hereon.

PropBank [Palmer et al., 2005] is both a corpus and an annotation schema in which verbal propositions and their numbered arguments are marked within the text. Roles are described through numbered arguments from `Arg0` up to (potentially) `Arg5` and through modifier markers denominated `ArgM`. An example is as follows:

(2.3) [`Arg0` Analysts] have been expecting [`Arg1` a pact...]

The usage of numbered arguments aims to address the lack of consensus concerning semantic roles. On the one hand, such approach avoids lack of annotation consistency, because numbered arguments are closer to syntax than semantic roles, resulting in a smaller semantic gap during analysis. On the other hand, arguments `Arg2` to `Arg5` are highly variable and overloaded, resulting in poor performance when attempting to map a given argument to a semantic role. A rough map is given below, but aside from clearly indicating overloading, this map is not true in a number of cases:

- `Arg0`: Agent, experiencer
- `Arg1`: Patient, theme
- `Arg2`: Benefactive, instrument, attribute, end state
- `Arg3`: Start point, benefactive, instrument, attribute
- `Arg4`: End point

As for modifier markers, some examples are directional markers (`ArgM-DIR`), locative markers (`ArgM-LOC`), manner markers (`ArgM-MNR`), temporal markers (`ArgM-TMP`), adverbials (`ArgM-ADV`), etc.

The second effort on SRL is FrameNet [Baker et al., 1998; Fillmore & Baker, 2001]. It is based on *frame semantics*, an extension of case theory in which the meaning of words is understood in terms of *semantic frames*. In the example 2.4, the concept of "*grilling*" is represented by a frame called `apply_heat`, which has as *frame elements* (FEs): a `cook` (person doing the cooking), `food` (the food that is to be cooked), `heating_instrument` (source of heat) and `container` (something to hold the food while cooking). Words that evoke `apply_heat`, such as "*grill*", "*bake*" and "*fry*", are called its *lexical units* (LUs).

(2.4) [$_{\texttt{cook}}$ They] grill [$_{\texttt{food}}$ their catches] [$_{\texttt{heating\_instrument}}$ on an open fire].

FrameNet defines frames and annotates sentences to show how FEs fit syntactically around the word that evokes the frame. It is closer to deeper semantics than PropBank, and consequently farther from the syntactic level.

The third effort is VerbNet [Schuler, 2005]. It organizes verbs according to Levin classes [Levin, 1993], which are hierarchically organized and capture generalizations about verb behavior. Members from a given class have common syntactic elements, semantic roles and syntactic frames. An example for class `Hit-18.1`, whose members include "*hit*" and "*kick*", is given in figure 2.1.

Class: `Hit-18.1`
Roles: Agent[+*int_control*] Patient[+*concrete*] Instrument[+*concrete*]
Frame 01: Basic Transitive
Syntax: Agent V Patient
Semantics: cause(Agent, E) manner(during(E),directedmotion,Agent)
!contact(during(E),Agent,Patient) manner(end(E),forceful,Agent)
contact(end(E),Agent,Patient)
Example: Paula hit the ball

FIGURE 2.1: Example of the VerbNet class `Hit-18.1`

Finally, SemLink [Palmer, 2009] is a project that aims to map PropBank, FrameNet, VerbNet and WordNet, combining information provided by these four different lexical resources.

2.1.2 Discourse Parsing

There are two notable efforts in discourse parsing, namely the RST Discourse Treebank and the Penn Discourse Treebank.

The RST Discourse Treebank [Carlson et al., 2001] is based on the Rhetorical Structure Theory (RST) [Mann & Thompson, 1988]. In this theory, the discourse structure of a text is represented as a tree, whose leaves are text fragments that represent *elementary discourse units* (EDUs), the minimal units of discourse. Each node of the tree then characterizes a *rhetorical relation* between two or more adjacent nodes. For example, the Contrast relation is illustrated in the sentence 2.5 below. A list of relation classes defined for the RST Discourse Treebank is given in appendix C.

(2.5) [Tiger's workers unionized,] [while Federal's never have.]

A novelty of RST is the concept of *nuclearity*. Rhetorical relations are characterized by a *nucleus* (head entity) and a *satellite* (tail entity), ultimately defining a *rhetorical structure tree*. There are, however, some multinuclear relations such as Contrast, for which there is no dominant entity.

The Penn Discourse Treebank (PDTB) [Miltsakaki et al., 2004] is a lexically-grounded approach to discourse parsing. It is built upon the theory by Webber & Joshi [1998], in which discourse connectives are considered to be predicates that take abstract objects as their arguments. An example in which the conjunction "*so*" is a Result discourse connective is given below in sentence 2.6. A list of classes, types and subtypes defined for the PDTB is given in appendix C.

(2.6) Even though critical, [$_{\texttt{Arg1}}$ it was just the kind of attention they were seeking]. So [$_{\texttt{Arg2}}$ they fired back at the Goldman Sachs objections in their own economics letter, "The BMC Report."]

Differences between the two approaches are then discussed. While RST is commited to building a discourse tree, PDTB is not. In fact, although text spans identified as arguments of a discourse connective never overlap, the same is not true for arguments of different connectives. In addition, PDTB works at the clausal level, whereas RST works at the subclausal level.

2.1.3 Concept Description Language

Concept Description Language (CDL) [Yokoi et al., 2005; Uchida et al., 2005; Zhu et al., 2005] is a language that abstracts the conceptual structure of text, proposed by the Institute of Semantic Computing of Japan (ISeC) and part of the Universal Networking Language (UNL) initiative. Its core defines a set of contextual semantic relation classes both at the intra- and inter-event levels, proposing a minimally sufficient set of relation classes that represents interlingual concept meaning. The complete list of CDL relation classes is given in appendix D.

An example of CDL-annotated sentence is given in figure 2.2. In this figure, the sentence "*John reported to Alice that he bought a computer yesterday*" is divided into six entities: "*John*", "*report*", "*Alice*", "*buy*", "*computer*" and "*yesterday*". The entities "*report*" and "*buy*" characterize events. The intra-event relations are: "*John*" $\xrightarrow{\texttt{agt}}$ "*report*", "*Alice*" $\xrightarrow{\texttt{gol}}$ "*report*", "*John*" $\xrightarrow{\texttt{agt}}$ "*buy*", "*computer*" $\xrightarrow{\texttt{obj}}$ "*buy*" and "*yesterday*" $\xrightarrow{\texttt{tim}}$ "*buy*". The only inter-event relation is: "*he bought a computer yesterday*" $\xrightarrow{\texttt{obj}}$ "*report*".

```
{#A01 Event tmp='past';
   {#B01 Event tmp='past';
      {#b01 buy;} {#b02 computer ral='def';} {#b03 yesterday;}
      [#b01 agt #John] [#b01 obj #b02] [#b01 tim #b03]
   }
   {#John John;} {#Alice Alice;} {#a01 report;}
   [#a01 agt #John] [#a01 gol #Alice] [#a01 obj #B01]
}
```

FIGURE 2.2: Example of the sentence "*John reported to Alice that he bought a computer yesterday*" annotated according to CDL

In terms of structure, CDL is closer to the original case theory at the intra-event level and to RST at the inter-event level. When comparing to SRL tasks, CDL uses well-defined semantic relations (unlike the numbered arguments from PropBank), and does not work directly with semantic frames (unlike FrameNet and VerbNet), although it can be connected to the frame-capable UNL Ontology [Uchida et al., 2005].

Finally, it should be noted that some relations such as `icl` (Inclusion or Kind-Of relations) are not contextual by nature. They can, nevertheless, be expressed within a context due to the property of *duality*, which states that the such relation classes can be described both by their members as well as by their contextual usage within natural language [Bollegala et al., 2009, 2010]. For example, the example between "*maltese*" and "*dog*" can be contextually stated in the sentence "*Maltese is a dog breed*".

2.1.4 Manual Annotation

Let a contextual semantic relation be described as $r : T \xrightarrow{C_k} H$, where H is the head entity, T is the tail entity and C_k is its relation class. Alternatively, it can be written as $r : T \rightarrow H$ for simplicity of representation. Let the set of entities within a text be written as E, such that $H, T \in E$; the set of relations as R, such that $r \in R$; the set of classes from a schema as C, such that $C_k \in C$; and the set of relations for a given class C_k be written as R_k, such that $R_k \subset R$. A directed hypergraph can be formed by entities E and relations R, which are respectively *hypernodes* and *edges* of the graph.

Two observations on simplifications of this hypergraph are then made:

- Simple graph representation: The hypergraph can be simplified to a simple graph if hierarchical nodes are connected by their roots. Ex.: In figure 2.2, "*bought*" can be considered the root for the clause "*he bought a computer yesterday*"
- Tree representation: The simple graph can be further simplified to a tree if coreference is disregarded. Ex.: In figure 2.2, the connection between "*John*" and "*he*" is a coreference, causing a loop in the graph

Although semantically poorer, the tree representation is easier to manually and automatically annotate. In addition, it is expected that this tree be similar to syntactic trees such as those generated by dependency parsing, given the hypothesis that semantic roles have a correlate construction at the syntactic level. Finally, transforming a tree representation into a hypergraph is possible through two steps: coreference resolution and a simple heuristic step which correlates noun and verb phrases from the syntactic tree to hierarchical nodes.

Given a syntactic tree, the annotation task for a schema such as CDL then consists of: (i) identifying entities E within the syntactic tree; (ii) identifying relations R among elements from E, forming the semantic tree; and (iii) assigning a class C_k for all relation in R.

First, for (i) identifying entities E within the syntactic tree, it is observed that the most granular entities are directly mappable to single words or expressions. By scanning through all words, it is possible to identify these granular entities in $\mathcal{O}(|W|)$, where $|W|$ is the number of words in the corpus. Less granular entities such as events can be easily identified once relations among entities are determined (transformation from tree to graph).

Second, for (ii) identifying relations R among elements from E, it is necessary to find the parent of each entity, which is possible in $\mathcal{O}(log|W|)$ per entity because of the tree structure when disconsidering co-reference.

Finally, for (iii) assigning a class C_k for all relations in R, it is necessary to classify each relation given some schema, which is possible in $\mathcal{O}(|C|)$, where $|C|$ is the number of relation classes.

The total annotation cost $\mathcal{C}$ based on equation (1.1) for this task for a set of languages L for a set of relation classes C is thus roughly estimated by

$$\mathcal{C}(L) = \sum_{L_i}^{L} \sum_{C_k}^{C} (T_S \cdot |W| + T_C \cdot |R_k|) \cdot A(L_i) \tag{2.1}$$

where T_S is the averate time to manually determine the graph structure of relations among entities for one relation (complexity of $\mathcal{O}(log|W|)$), and T_C is the average time to manually determine the relation class for one relation (complexity of $\mathcal{O}(|C|)$).

If it is possible to determine heuristics for automatically extracting the relation graph from syntactic trees, T_S can be decreased to 0. In this case, when disconsidering the cost of creation of these heuristics, the equation (2.1) becomes

$$\mathcal{C}(L) = \sum_{L_i}^{L} \sum_{C_k}^{C} T_C \cdot |R_k| \cdot A(L_i) \tag{2.2}$$

2.2 Semantic Analysis

2.2.1 Problem Statement

Similarly to manual annotation of contextual semantic relations, the analysis task consists of two main problems: identification and classification.

Identification regards the extraction of candidate relations, by identifying both entities E within the text and the existence of relations R among them, without assigning relation classes C_k to the relations. The main difficulty for this step is to translate a syntactic

parse tree into a semantic relation tree by analyzing word forms, especially connecting words such as prepositions and conjunctions.

Classification regards the assignment of classes C_k to each relation. The main difficulties for this step is to obtain linguistic features that are able to properly describe relations, and to abstract these features into a proper model.

2.2.2 Corpora and Systems

The CoNLL-2004 and CoNLL-2005 Shared Tasks [Carreras & Màrquez, 2004, 2005] are two tasks for which many systems attempted to assign PropBank semantic roles to texts. For the 2005 task, sections 02-21 from the Wall Street Journal (WSJ) dataset were used as training set and section 24 as development set. The WSJ corpus is part of the Penn Treebank II [Marcus et al., 1993]. For testing, both section 23 of the WSJ and three sections from Brown Corpus [Kučera & Francis, 1967] were used. Detailed information on each corpus is given on tables 2.1 and 2.2.

Table 2.1: Statistics on the CoNLL-2005 ST corpus: Counts

	Train	Dev	Test (WSJ)	Test (Brown)
Sentences	39,832	1,346	2,416	426
Tokens	950,028	32,853	56,684	7,159
Propositions	90,750	3,248	5,267	804
Verbs	3,101	860	982	351
Arguments	239,858	8,346	14,077	2,177

The texts were preprocessed using the POS tagger SVMTool[1] [Giménez & Màrquez, 2003], a phrase chunker and clause parser [Carreras & Màrquez, 2003], the full parsers Collins [Collins, 2003] and Charniak [Charniak, 2000], and a named entity recognizer [Chieu & Ng, 2003]. The participating systems, together with their methods and linguistic features, are described in table 2.3.

As for CDL, there are two main annotated corpora: one created by annotating existing texts, and the other created anew for the CDL specification.

The WikiCDL corpus is composed of manually annotated texts from Wikipedia for the following nine articles: Disease, Gene, Knowledge Management, Medicine, Science, Semantic Web, Technology, Virus and Web Search Engine. The annotated articles went through a process of reviewing, eliminating instances that would be ambiguous for human annotators. This dataset contains examples of 39 CDL relation classes, with a total of 12,277 instances and an average of 314.79 instances per class.

[1] `http://www.lsi.upc.edu/~nlp/SVMTool/`

TABLE 2.2: Statistics on the CoNLL-2005 ST corpus: Core arguments and adjuncts

	Train	Dev	Test (WSJ)	Test (Brown)
A0	61,440	2,081	3,563	566
A1	84,917	2,994	4,927	676
A2	19,926	673	1,110	147
A3	3,389	114	173	12
A4	2,703	65	102	15
A5	68	2	5	0
AM-ADV	8,210	279	506	143
AM-CAU	1,208	45	73	8
AM-DIR	1,144	36	85	53
AM-DIS	4,890	202	320	22
AM-EXT	628	28	32	5
AM-LOC	5,907	194	363	85
AM-MNR	6,358	242	344	110
AM-MOD	9,181	317	551	91
AM-NEG	3,225	104	230	50
AM-PNC	2,289	81	115	17
AM-TMP	16,346	601	1,087	112

The SpecCDL corpus is composed of relations instances available as examples in the CDL specification. For instance, for the possession relation (`pos`), the example stated in the specification is the relation between "*John*" and "*dog*", in the sentence "*John's dog*". There are relations for all 46 classes, and each class has at most 6 examples, being 2.41 the average number of examples per class.

The extraction of CDL relations consisted of heuristic identification and supervised classification steps [Yan et al., 2008; Yan, 2010]. For the heuristic identification step, a set of hand-made rules was applied to the dependency tree of each input sentence in order to extract candidate head and tail entities. A simple post-processing step then corrected the boundaries within which the dependency tree did not present correct relationships. For the supervised classification step, linguistic features were first extracted from candidate relations. The following features were used:

- Morphology tag: Information on word form extracted by the Connexor Parser[2], which defines 70 tags. For example, five tags are assignable for nouns, namely `N` (noun), `SG` (singular), `PL` (plural), `NOM` (nominative) and `GEN` (genitive)
- Syntactic tag: Information on syntax function of words also extracted by the Connexor Parser[2], which defines 40 tags. For example, `%NH` (nominal head) and `%>N` (determiner or premodifier of a noun) are surface syntax tags, and `@SUB` (subject) and `@F-SUB` (formal subject) are syntactic function tags

[2] `http://www.connexor.eu/technology/machinese/`

TABLE 2.3: CoNLL-2005 ST participating systems

Name	Method	Features				
		cha	col	pos	chk	ne
Punyakanok	SNoW	x	x	x	—	x
Haghighi	MaxEnt	x	—	—	—	—
Màrquez	AdaBoost	x	—	x	—	x
Pradhan	SVM	x	x	x	x	x
Surdeanu	AdaBoost	x	—	—	—	x
Tsai	MaxEnt,SVM	x	—	x	—	x
Che	MaxEnt	x	—	—	—	x
Moschitti	SVM	x	—	—	—	—
Tjong	MaxEnt,SVM,MBL	x	—	—	—	x
Yi	MaxEnt	x	—	—	—	—
Ozgencil	SVM	x	—	—	—	—
Johansson	RVM	x	—	x	—	x
Cohn	Tree-CRF	—	x	—	—	—
Park	MaxEnt	x	—	—	—	—
Mitsumori	SVM	x	—	x	x	x
Venkatapathy	MaxEnt	—	x	—	—	x
Ponzetto	DT	—	x	x	—	x
Lin	CPM	x	—	—	—	—
Sutton	MaxEnt	—	—	—	—	—

Methods are Sparse Network of Winnows (SNoW), Support Vector Machines (SVM), Relevance Vector Machines (RVM), Conditional Random Fields (CRF), Decision Trees (DT), Combinatorial Pattern Matching (CPM) and Memory-based Learning (MBL). Linguistic features are Charniak parsing (cha), Collins parsing (col), SVMTool parsing (pos), chunking (chk) and named entities (ne).

- Dependency shortest path: Information on the shortest path [Bunescu & Mooney, 2005] between head and tail entities of the dependency tree. In addition, the dependency functions of the entity pair are also considered by using the dependency labels of the main word of both head and tail entities

- Word sense: Lexical information on synonymy and hypernymy of words from WordNet (see section 1.1.2)

- Word behavior: Information extracted from the UNLKB [Uchida & Zhu, 2002], which is a lexicon that organizes words in a hierarchical structure based on the semantic behavior. For example, one of the word behaviors of "*give*" is *give(agt >thing, obj >thing)*

Given this feature set, feature vectors were created for training data instances. They were then used for training and classification using a Support Vector Machine (SVM) [Joachims, 1998]. For this task, the SVM-Light package[3] was used.

[3] `http://svmlight.joachims.org/`

2.2.3 Experiments and Results

For the CoNLL-2005 ST, systems stated in table 2.3 attempted to classify semantic roles the WSJ and Brown corpora. Classification results in terms of precision, recall and F-value were obtained as indicated in table 2.4. The best system for each corpus is highlighted.

TABLE 2.4: CoNLL-2005 ST results

Name	WSJ			Brown		
	$P(\%)$	$R(\%)$	$F(\%)$	$P(\%)$	$R(\%)$	$F(\%)$
Punyakanok	**82.28**	**76.78**	**79.44**	73.38	62.93	67.75
Pradhan	82.95	74.75	78.63	**74.49**	**63.30**	**68.44**
Haghighi	79.54	77.39	78.45	70.24	65.37	67.71
Màrquez	79.55	76.45	77.97	70.79	64.35	67.42
Pradhan	81.97	73.27	77.37	73.73	61.51	67.07
Surdeanu	80.32	72.95	76.46	72.41	59.67	65.42
Tsai	82.77	70.90	76.38	73.21	59.49	65.64
Che	80.48	72.79	76.44	71.13	59.99	65.09
Moschitti	76.55	75.24	75.89	65.92	61.83	63.81
Tjong	79.03	72.03	75.37	70.45	60.13	64.88
Yi	77.51	72.97	75.17	67.88	59.03	63.14
Ozgencil	74.66	74.21	74.44	65.52	62.93	64.20
Johansson	75.46	73.18	74.30	65.17	60.59	62.79
Cohn	75.81	70.58	73.10	67.63	60.08	63.63
Park	74.69	70.78	72.68	64.58	60.31	62.38
Mitsumori	74.15	68.25	71.08	63.24	54.20	58.37
Venkatapathy	73.76	65.52	69.40	65.25	55.72	60.11
Ponzetto	75.05	64.81	69.56	66.69	52.14	58.52
Lin	71.49	64.67	67.91	65.75	52.82	58.58
Sutton	68.57	64.99	66.73	62.91	54.85	58.60
Baseline	51.13	29.16	37.14	62.66	33.07	43.30

As for CDL, experiments were held using different combinations for the feature set, namely: (`A`) only the morphological and syntactical features, (`B`) only the dependency features, (`C`) only the lexical-semantic features, (`D`) the morphological, syntactical and dependency features, and (`E`) all features. Classification results are stated in table 2.5. From these five different settings, the one that rendered the highest preliminary values of precision, recall and F-value is the last one.

The results for the semantic analysis task using the WikiCDL corpus and ten-fold cross-validation are shown in table 2.6. The numbers indicate that values for the supervised classification were satisfactory, but these high values may be explained by the usage of word behavior features with frame capabilities. On the other hand, heuristic identification produced fairly low results, which could be improved by integrating information from different levels of natural language processing.

TABLE 2.5: Results of semantic relation classification for CDL (per feature set)

Feature Set	P(%)	R(%)	F(%)
`A`: Morphosyntactic	79.33	85.78	82.43
`B`: Dependency	83.62	83.56	83.59
`C`: Lexico-semantic	73.49	81.63	77.35
`D`: `A` + `B`	85.63	85.91	85.77
`E`: `C` + `D`	**86.35**	**87.43**	**86.89**

TABLE 2.6: Results of semantic relation analysis for CDL (per step)

Step	P(%)	R(%)	F(%)
Relation Identification (RI)	62.65	68.33	65.37
Relation Classification (RC)	86.35	87.43	86.89
RI + RC	51.62	57.94	54.60

It is important to notice, however, that the relation classification task gives equal weights for each relation instance. As a result, it is understood that the classification is overall satisfactory, but it is not necessarily true that results for each individual class are at acceptable levels. Errors in classes with few instances go unnoticed with standard evaluation metrics.

2.3 Alternative Semantic Analysis

2.3.1 Problem Statement

In supervised classification, if there is only a small number of instances of a class in training data, it is expected that the classifier will not have enough information from which to abstract a reliable classification model, in a problem denominated *incomplete feature space*. Features that occur only in the testing data are overseen, which in turn negatively affects performance. Different ways to address this problem for semantic relations have been attempted, two of which are detailed hereon.

One way is to use *bootstrapped set expansion*. Li et al. [2011] used the dual property of uncontextualized relations, which states that a semantic relation between nominals is defined by both the lexical contexts (intensional definition) and by the entity pairs (extensional definition) [Bollegala et al., 2010]. For example, the Capital-Country can be described by either the set of contexts such as "*X is capital of Y*" and "*Y's capital X*", or by the set of entity pairs such as "*Tokyo*" and "*Japan*". Given an initial set of contexts, the aforementioned work uses web search engine queries to extract more entity pairs, which are added to the initial set of entity pairs. Likewise, the initial set of entity

pairs is used for producing more contexts to be added to the initial set of contexts. After several iterations of a bootstrapped process [Agichtein et al., 2001], both sets grow considerably.

The other way is to use *feature vector extension*. Hernault et al. [2010] used correlation among features in order to enrich the feature space for discourse relations in a semi-supervised manner. Under this approach, features from both labeled and unlabeled data are extracted, and correlation among them is calculated. This correlation is used for adding new features to the training vectors in order to account for information that would not be present otherwise.

For the domain of contextual semantic relations, the problem of *class imbalance* [Japkowicz & Stephen, 2002] is observed. Class imbalance states that the distribution of relations is not homogeneous, with some classes having very few instances. In the WikiCDL corpus, for example, twelve classes presented more than 0 and fewer than 10 instances. The observed frequencies are shown in table 2.7.

TABLE 2.7: Frequencies for CDL relation classes in the WikiCDL corpus

Class	Freq	Class	Freq	Class	Freq	Class	Freq
agt	874	equ	31	obj	3339	qua	326
and	1185	fmt	9	opl	4	rsn	40
aoj	2399	frm	20	or	231	scn	54
bas	17	gol	120	per	2	seq	1
ben	17	icl	0	plc	186	shd	0
cag	2	ins	3	plf	0	src	65
cao	0	int	0	lt	1	tim	124
cnt	45	iof	38	pof	14	tmf	8
cob	0	man	863	pos	165	tmt	4
con	8	met	28	ptn	20	to	37
coo	0	mod	1807	pur	164	via	4
dur	37	nam	9				

Infrequent relations cause problems for the machine learner, as previously mentioned. Because of class imbalance, it may not be assumed that a larger annotated corpus necessarily results in enough training examples. In addition, the manual crafting of examples without using a real world corpus (ex.: SpecCDL) may lead to a dataset that does not reflect application data.

This work follows the idea by Li et al. [2011] and proposes bootstrapped set expansion in order to extract new training instances from a unlabelled corpus such as the web. Feature vector expansion is not applicable because correlation among morphosyntactic features is not assumed to improve classification in this case. For example, NN (POS tag for singular nouns) and "*financial institution*" (word sense of "*bank*") are clearly

correlated, but this correlation is already expressed by existing features. One challenge of using set expansion is that unlike uncontextualized relations, contextualized ones do not possess the dual property. Their structures, however, are similar to syntax trees. By using syntactical pattern matching on web query results, it is expected that the amount of noise be greater, requiring a filter which is able to perform even under incomplete feature spaces.

Active learning has been used for contextual relations in a work by Roth & Small [2006] on the CoNLL-2004 Shared Task with candidate relations annotated, i.e. assuming the identification step done in an unlabeled corpus. The problem when using a PropBank-based corpus is that class imbalance is not observed, as this schema has only a few relation classes in order to decrease annotation inconsistencies. Co-training has also been attempted [He & Gildea, 2006], but without any significant improvements due to imbalanced views, among other reasons.

The proposed research considers the entire iterative process, studying the identification of infrequent classes using web searching and syntactic pattern matching. The usage of web querying addresses imbalance, at the expense of increased amount of noise. As a result, it proposes a novel distance-based weak classifier with confidence-based output which focuses on classifying true positives while addressing the incomplete feature space problem.

Finally, this work proposes semi-automatic bootstrapped set expansion by adding a manual annotation step to the process. While active learning approaches aim to sample a minimally descriptive training dataset, this work attempts to obtain a large complete dataset with minimal effort using annotation-assistance, since it has been observed that an increase in the size of training data improves overall classification [Carreras & Màrquez, 2005], especially when word sense and word behavior features are important descriptors for the model.

2.3.2 Bootstrapped Set Expansion

2.3.2.1 Overview

Let R_0 be the original training data from the semantic analysis task, consisting of infrequent relation classes C_{k^*}. Bootstrapped set expansion is the process that extracts new instances that are relevant to classification from a large unannotated corpus such as the web, generating sets $R_1, R_2, \ldots, R_n$ after successive iterations of the process. The feature space coverage for these expanded sets is increased by adding instances with new or recombined features.

The overview of the proposed approach is illustrated in figure 2.3. For each infrequent class C_{k^*}, for each iteration i of the bootstrapped process, it uses all relations in R_{i,k^*} (set of relations of infrequent class C_{k^*} for the current iteration) for web querying, retrieving sentences that may contain a new instance. Syntactic pattern matching is then used in order to extract candidate relations, which are evaluated as relevant for C_{k^*} through the two steps of filtering, an automatic and a manual one. If a relation passes both filters, it is added to C_{k^*} and is used as seed for the next iteration.

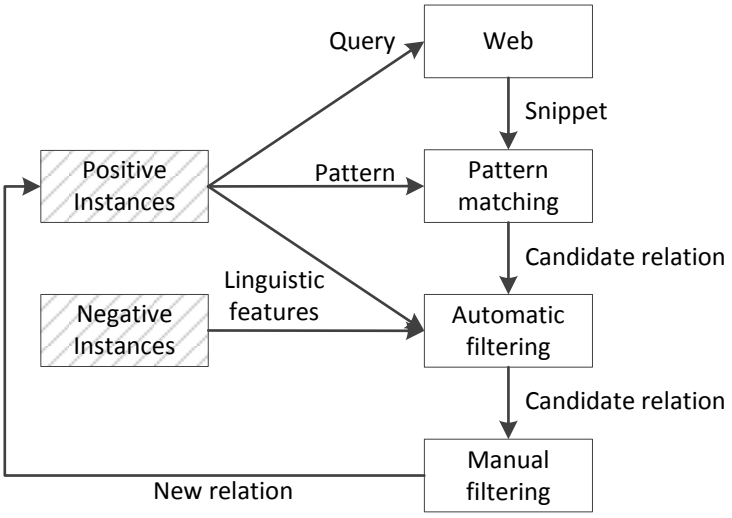

FIGURE 2.3: Overview of bootstrapped set expansion for one relation class

For each class C_{k^*}, positive instances R^+ and negative instances R^- are defined. Positive instances are elements from the set R_{i,k^*}. Negative instances are elements from relation classes deemed similar to C_{k^*}, i.e. relations which are difficult to differentiate. For example, relations such as `met` (method) and `ins` (instrument) may not be trivial to differentiate because they belong to the same macro-group Instrument. Table 2.8 illustrates sentences from `met` and `ins` classes, and appendix D describes macro-groups for CDL.

TABLE 2.8: Examples of positive and negative instances for the Method class

Set	Relation	Sentence
R^+	"*product*" $\xrightarrow{\texttt{met}}$ "*categorize*"	"*... categorizes genes by (...) products*"
	"*self-assembly*" $\xrightarrow{\texttt{met}}$ "*create*"	"*... creating (...) through self-assembly*"
	"*experiment*" $\xrightarrow{\texttt{met}}$ "*disprove*"	"*... partially disproved by experiment*"
R^-	"*eye*" $\xrightarrow{\texttt{ins}}$ "*see*"	"*... which can often be seen with the unaided eye...*"
	"*hammerstone*" $\xrightarrow{\texttt{ins}}$ "*strike*"	"*... was struck with a hammerstone*"
	"*microscope*" $\xrightarrow{\texttt{ins}}$ "*see*"	"*... seen directly with a light microscope*"

Aside from the amount of noise, one of the main challenges of this approach concerns the automatic filtering step, which is used in order to decrease the number of manually

annotated instances. The small size of the seed set R_0 affects supervised learners used for filtering because of the incomplete feature space. In this work, feature distance metrics are used, since they do not generate sparsity and do not penalize the classifier when there are features inexistent in the training set. When using such distance metrics, properly accounting for the importance of features of different types becomes a non-trivial matter. While in Yan et al. [2008] this is done automatically by the SVM, such unified view is also necessary for the distance-based approach. This is achieved by using multiview distance matrices.

2.3.2.2 Web Search Querying

The web search step is carried in order to find sentences which may contain relation instances that are somewhat similar to any of the positive relation instances, hence being relevant to the R^+ set. For each positive relation in R^+, two queries are formed: one explicitly states the head entity and substitutes the tail for a wildcard, and the other explicitly states the tail and substitutes the head for a wildcard. In both cases, relevant morphosyntactic elements such as prepositions within the shortest path between head and tail are preserved, and all words are ordered according to their original positions within the text. An example of sentence in 2.7 with generated queries 2.7a and 2.7b is given below:

(2.7) "*product*" $\xrightarrow{\texttt{met}}$ "*categorize*" in "*... categorizes genes by (...) products*"

a. "*categorize by **"

b. "** by products*"

By omitting one of head or tail entities, this step returns search results that might contain new head-tail pairs. A search result is illustrated in table 2.9. Using the above example, query 2.7a returns web page descriptions that contain expressions such as "*categorize by color*" and "*categorize by name*", whereas 2.7b returns several sentences with the word "*by-product*" in them, which is clearly not in the `met` class. In next iterations, results should include instances such as "*shop by color*" and "*search by name*", enriching the feature space while increasing training dataset size.

It is observed that queries such as "*categorize * products*", which explicitly state both head and tail entities but substitute the morphosyntactic elements with a wildcard, are not used. Although such query might generate new syntactic structures, the amount of noise does not justify its usage. As an example, not one of the first 100 results for query "*categorize * products*" belonged to the `met` class. In addition, it is not trivial to

TABLE 2.9: Example of web query search result

Field	Content
Query	"*categorize by **"*
URL	*http://www.merriam-webster.com/dictionary/categorized*
Title	*Categorized - Definition and More from the Free Merriam ...*
Description	*Birds are categorized by type in this field guide. First known use of categorize. 1705. Related to categorize. Synonyms: assort, break down, classify, class, codify, compartment, ...*

evaluate the class for sentences such as "*categorize its products*" at this point, since this is the ultimate objective of the analysis task for semantic relations.

2.3.2.3 Pattern Matching

Given the correlation between syntax and contextual semantic relations, the syntactic structure of the sentence is used in order to create patterns which identify candidate relations. Two types of patterns are investigated herein, using shortest paths for phrase structures and for dependency trees.

For phrase structures, patterns include the following information: (1) POS tags of head and tail entities; (2) phrasal categories of intermediate nodes; (3) indication of the highest node in the tree; and (4) order of entities in the sentence. Moreover, if there are any prepositions, conjunctions, etc. within the shortest path, they should also be explicitly stated. An example is given (double brackets indicate the highest node):

(2.8) [head] [VB(Z)] [[VP]] [PP_by] [NP] [NN(S)] [tail]

For dependency trees, patterns are more straight forward, as prepositions and conjunctions are explicitly stated for collapsed dependencies. An example is given:

(2.9) [[head]] [prep_by] [tail]

Words and wildcards can also be used as elements within the pattern. In the example below, dependency pattern 2.10a matches the sentence in 2.10b, extracting the relation "*black*"→"*prefer*".

(2.10) a. [[head]] [acomp] [*] [prep_to] [tail]

b. "*black*"→"*prefer*" in "*I prefer blue to black*"
prefer [acomp] blue [prep to] black

Given their natures, it is expected that phrase structures generate results with more quantity and more noise (higher growth), whereas using dependencies generate results with less quantity and more quality (higher precision). This is because the first explain syntactic relations based only on prepositions or conjunctions, while the latter explain based on the syntactic role as well.

The candidate relations extracted from pattern matching are selected using automatic and manual filtering steps. In section 2.3.2.4, feature distance is introduced in the context of semantic relation analysis. In section 2.3.2.5, these distances are used for building a multiview distance matrix. Finally, in section 2.3.2.6, automatic and manual filtering, as well as an overview of the bootstrapped process, are detailed.

2.3.2.4 Feature Distance Metrics

Features are used in order to linguistically describe semantic relations. Feature types used in this work as stated below. For WordNet senses, disambiguation is not performed (i.e. all senses are considered).

- POS tag of head entity
- POS tag of tail entity
- Phrase structure shortest path
- Dependency tree shortest path
- Named entity tag of head entity
- Named entity of tail entity
- WordNet sense of head entity
- WordNet sense of tail entity

Given the difficulty of feature vectors to account for unknown features, feature distances are used herein. For each feature type f_k, a distance function δ_k is determined. This function takes two relations as input, and returns a value in the range $[0, 1]$, where 0 is when two features present the same feature value for type f_k. The distance functions used are:

- POS tags (head and tail): Pre-defined distances among POS tags

- Phrase structure and dependency tree shortest paths: Normalized generalized Levenshtein distance
- Named entity tags (head and tail): Binary value
- WordNet senses (head and tail): Distance on hierarchy tree

For POS tag features, since there is a limited number of possible tags, a table of predefined distances was created. It is illustrated in appendix A. As a general rule, the more similar the tags are (as would be the case of NNP and NNS), the lower the distance, since similar word classes possess tag prefix.

For phrase structure and dependency trees shortest paths, a distance measure for arrays is necessary. As a result, a modification of the Levenshtein edit distance is used. It counts the minimum number of edits to transform one sequence into another, using only insertion, deletion and substitution operations. Although the Levenshtein distance was originally used for strings, it can also be used for general purpose arrays. The distance calculation is shown in algorithm 1. Notice that the result for the algorithm is later normalized, being divided by the length of the longest array.

Algorithm 1 Modified Levenshtein distance

```
1:  function LevenshteinDistance(m-dimensional array s, n-dimensional array t)
2:      d ← new m × n matrix
3:      for i = 0 → m do
4:          d[i, 0] ← i
5:      for j = 0 → n do
6:          d[0, j] ← j
7:      for j = 1 → n do
8:          for i = 1 → m do
9:              if s[i] = t[j] then d[i, j] ← d[i − 1, j − 1]
10:             else d[i, j] ← min(d[i − 1, j] + 1, d[i, j − 1] + 1, d[i − 1, j − 1] + 1)
11:     return d[m, n]
```

For named entity tags, the distance metric used is a binary one. In other words, the distance function outputs 0 if the two tags are the same, or 1 otherwise.

Finally, the distance function for word senses is a simple one based on the lexical hierarchy of WordNet. Given two senses s_1 and s_2, cp the common parent between them, r the root of the tree, and $d(n_i, n_j)$ the number of edges between two nodes n_i and n_j, the distance is calculated as shown in equation (2.3).

$$\delta_{wn} = \frac{min(d(s_1, cp), d(s_2, cp))}{d(cp, r) + min(d(s_1, cp), d(s_2, cp))} \tag{2.3}$$

An example of feature types, features and distance values is given in table 2.10 for the relation "*product*"→"*categorize*" in sentence "*... categorizes genes by (...) products*" and the relation "*product*"→"*train*" in sentence "*certification training by product*".

TABLE 2.10: Example of feature distance metrics

f_k	Features		δ_k
	categorize→product	*train→product*	
Head POS	VBZ	NN	1.0
Tail POS	NNS	NN	0.3
Phrase	[head] [[VP]] [PP_by] [NP] [tail]	[head] [NP] [[FRAG]] [PP_by] [NP] [tail]	0.5
Dep	[[head]] [prep_by] [tail]	[[head]] [dep] [*] [pobj] [tail]	1.0
Head NE	NONE	NONE	0.0
Tail NE	NONE	NONE	0.0
Head WN	categorize#1	educate#3, etc.	1.0
Tail WN	merchandise#1, etc.	merchandise#1, etc.	0.0

Features of type f_k and distance values δ_k are given for relation "*categorize*"→"*product*" in sentence "*... categorizes genes by (...) products*" and "*train*"→"*product*"

2.3.2.5 Multiview Distance Matrices

One problem of using feature distances is how to deal with features of different types. It is necessary to obtain a unique representation of a model for a domain that is inherently multiview. *Multiview distance matrices* are then used for this matter.

First, given a set of relation instances $R' = R^+ \cup R^-$ and its size $n = |R'|$, a *single view distance matrix* D_k of a feature type f_k is an $n \times n$ matrix whose elements $d^k_{i,j} \in D_k$ are given by

$$d^k_{i,j} = \delta_k(r'_i, r'_j), \quad r'_i, r'_j \in R' \tag{2.4}$$

The multiview distance $\mathcal{D}$ is then defined as an $n \times n$ matrix which is calculated by a kernel that properly weights all distance matrices D_k. Assuming a linear kernel, the multiview distance matrix is given by the following equation

$$\mathcal{D} = \beta_0 + \sum_{i=1}^{k} \beta_i \cdot D_i \tag{2.5}$$

Hence, the single view matrix provides distances among all relations to be considered for a given feature type, whereas the multiview matrix provides a combined value for distances among all relations for different feature types. The constant term β_0 can be included in the sum by defining D_0 as a matrix of ones. As a result, the previous equation can be rewritten as follows in order to simplify notation:

$$\mathcal{D} = \sum_{i=0}^{k} \beta_i \cdot D_i \tag{2.6}$$

The β coefficients represent the weights of each feature type for the representation of a given relation class. Defining an *expected multiview distance matrix* $\mathcal{D}'$ as values that $\mathcal{D}$ would have in an ideal situation, for which $\mathcal{D}'_{i,j} = 0$ if both r'_i and r'_j are positive or $\mathcal{D}'_{i,j} = 1$ otherwise, the optimal β values are given by minimizing the quadratic error $(\mathcal{D}' - \mathcal{D})^2$:

$$\frac{\partial}{\partial \beta_x} \left[\mathcal{D}' - \sum_{i=0}^{k} \beta_i D_i \right]^2 = 0, \quad x = 0, \ldots, k \tag{2.7}$$

The task of minimizing the error in equation 2.7 is thus equivalent to finding β coefficients for the system of equations indicated below. The values of β can be easily achieved using least squares multiple regression algorithms for the expected matrix $\mathcal{D}'$.

$$\mathcal{D}' = \sum_{i=0}^{k} \beta_i \cdot D_i \tag{2.8}$$

Least squares multiple regression uses single vector decomposition (SVD), which has a complexity of $O(n^2(k+1)^2)$, with n^2 being equivalent to the number of equations in the system. Since k is low and constant and n is restricted to the small size of the seed set, the quadratic performance should not be an issue in this case. Moreover, it is also possible to decrease the complexity of n^2 by ignoring elements r'_i and r'_j that are not useful for the task:

- If $D_{i,j}$ is used, $D_{j,i}$ may be discarded because distance functions are symmetric
- If both r'_i and r'_j belong to negative classes, $D_{i,j}$ may be discarded because the task is concerned with classifying the positive class

After calculating the β values, it is possible to calculate the multiview distance $\mathcal{D}$ from equation (2.6). By determining $\mathcal{D}$, it becomes possible to analyze the initial seed data, feature choice and intermediate steps of the process. In other words, this framework allows to:

- Analyze how the initial dataset responds to chosen features
- Analyze how well the chosen features characterize the initial dataset

- Predict the quality of the classification by measuring how well the multiview distance matrix separates positive and negative instances

This deep analysis of the initial dataset is definitely important for the bootstrapped process of infrequent relation classes, because the quality of the final output is directly dependent on the quality of the initial seed and because it becomes the training data of the analysis task as presented in section 2.2.

2.3.2.6 Filtering and Bootstrapping

Given a candidate relation extracted from web querying and pattern matching, automatic filtering regards finding the group of instances in the seed set which is closest to the candidate according to the distance metrics defined in the previous section. If the candidate is closest to a positive group, then it is regarded as a positive instance; otherwise, it is regarded as negative. Automatic filtering is then divided into two steps: clustering and classification.

Clustering defines many positive and negative groups. The idea of calculating distance to a group of instances is more interesting than calculating distance to a single instance because of bias, and is more interesting than calculating distance to the whole set of instances because there is more than one way to linguistically describe each relation class.

After a multiview distance matrix $\mathcal{D}$ is calculated, as detailed in previous section 2.3.2.5, this matrix is clustered by spectral clustering [Shi & Malik, 2000; Kannan et al., 2004], an algorithm which uses the spectrum of a similarity matrix in order to perform dimensionality reduction, and which is commonly used for image segmentation. Aside from $\mathcal{D}$, it has as input a parameter α^*, which controls the recursive partitioning. It is found by grid search, as its optimal value is the one that: (1) separates relation instances from different relation classes; (2) better separates instances from the same class if they are similar; and (3) better groups similar instances from the same class.

Classification then defines the assignment of a candidate instance to an existing group. The relation classification is distance-based, as mentioned previously. Given a candidate relation instance r_i and a cluster C obtained from the spectral clustering, the distance between them is a function of the distance to all cluster members. For example, if the function used is the average distance to all members, distance is given by equation (2.9).

$$\delta(r_{test}, C) = \frac{1}{|C|} \sum_{r_i}^{C} \delta(r_{test}, r_i) \tag{2.9}$$

The confidence measure θ' ranges from -1 to $+1$, with values closer to -1 indicating that r_{test} is more confidently classified as negative, values closer to $+1$ as positive, and values closer to 0 indicating that the classification is inconclusive. For set expansion, θ' is then compared against a pre-defined threshold θ, and test relations are carried for the next step of the bootstrapped process if and only if $\theta' \leq \theta$ holds true; they are otherwise discarded. This aims to increase precision of the classification. This approach may alternatively be used under an active learning framework by considering values closest to the margin (closest to 0).

The candidate instances that pass automatic filtering step have to be manually filtered in order to avoid error propagation in future iterations. Given a contextualized head and tail entity pair, the manual step consists of deciding if this pair belongs to the positive class under consideration. If it is decided to be positive, it is added to R_{i,k^*}, being used in next iterations as well as in the semantic analysis task. For the manual annotation, the confidently classified instances are ranked according to the confidence value θ'.

The iterative bootstrapped process can be run repeatedly until the dataset has improved considerably in quantity and quality. It is expected that the set of positive relations is larger than the original one, with new and recombined features.

2.3.3 Experiments and Results

2.3.3.1 Experimental Setting

Two different evaluations are performed: (i) evaluation of the expanded sets obtained through bootstrapped; and (ii) evaluation of the effects of expanded sets on the semantic analysis task.

For (i) evaluation of the expanded sets obtained through bootstrapped, the quality of the expanded datasets produced by the bootstrapped process is compared when using the proposed method (spectral clustering and distance-based classification) and when using an SVM baseline. In addition, phrase structure-based and dependency-based pattern matching approaches, as indicated in section 2.3.2.3, are also compared. For the experiments, the initial seed is composed of relation classes with at most ten instances randomly chosen from the WikiCDL corpus.

For the SVM baseline, for each relation instance r_i, a d-dimensional feature vector $v^i = [v^i_1, \ldots, v^i_d]$, $v^i_j \in \mathbb{R}$ is extracted. Each feature vector element v^i_j indicates the existence or inexistence of the feature v_j for the given instance r_i. The SVM input then becomes the following:

$$<+1|-1> : <v_1^i> <v_2^i> <v_3^i> \ldots <v_d^i>$$

The confidence value for this case is the sigmoidal probabilistic output, originally proposed by Platt [1999] and further extended by Lin et al. [2007]. This method maps the output of the classifier to the positive class posterior probability by applying the sigmoid function in equation (2.10), where f_i is the output of the SVM for x_i, $p_i = p(x_i)$ is the posterior class probability for x_i, and A and B are parameters found by minimizing the negative log-likelihood function of the training data. The resulting confidence value θ' is multiplied by -1 if the result of the classification is a negative cluster.

$$p_i = \frac{1}{1 + e^{Af_i+B}} \tag{2.10}$$

In this evaluation, standard metrics such as accuracy, precision, recall and F-value may not be completely suited for evaluation of tasks concerning infrequent relation classes, since the results are biased towards frequent classes. Macro-average metrics have been used in such situations [Hernault et al., 2010] because they weight classes equally, unlike standard micro-average metrics, which weight instances equally. As a result, macro-average gives more weight to classes that occur less frequently. For example, if the frequent class R_A got 99 correct and 0 incorrect instances and the infrequent class R_B got 0 correct and 1 incorrect instances, then micro-average accuracy would be 0.99 whereas macro-average accuracy would be 0.5.

It is observed that macro-average recall is not relevant for set expansion, since false negatives may be ignored without further losses to the process. As a result, only macro-average accuracy and precision are considered.

For (ii) evaluation of the effects of expanded sets on the semantic analysis task, the semantic analysis task by Yan et al. [2008] is compared under two settings: using non-expanded training sets and using expanded ones. In this case, both micro-average and macro-average metrics are used to measure precision, recall and F-value after different number of iterations of the process.

The training data in this case is composed of manually generated CDL relations from SpecCDL, with an average of 4.54 instances per each of the 46 classes, and the testing data consists of all semantic relations found in the WikiCDL, accounting for overall 12,277 instances from 39 CDL relations. The bootstrapped process for this task includes the manual checking step in order to avoid error propagation, and two iterations of the process were run for each relation class.

2.3.3.2 Evaluation of Set Expansion

For the evaluation of set expansion, macro-average accuracy and precision are used for comparing different methods (proposed vs SVM+Feature vectors) and syntactic patterns (phrase structure vs dependency). Macro-average metrics are calculated only for relation instances that were confidently classified. In other words, given a pre-defined minimum confidence value θ, relation instances with $\theta' < \theta$ are ignored.

Table 2.11 presents macro-average precision and accuracy values for the optimal threshold value obtained, which was determined experimentally. The resulting expanded instances were manually checked in this case.

TABLE 2.11: Macro-average performance of set expansion

Method	Phrase Structure		Dependency	
	MA-Acc	*MA-P*	*MA-Acc*	*MA-P*
Feature distances+Regression (proposed)	0.62	0.30	0.50	0.45
Feature vectors+SVM	0.59	0.18	0.45	0.25

The SVM with feature vectors baseline was outperformed in this task because the few number of training instances and the sparsity of the vector produced an incomplete classification model for the SVM.

In addition, when comparing different pattern matching, macro-average precision is higher when using dependency, but accuracy is higher when using phrase structure. This is an indication that phrase structure matching extracts more candidate instances, allowing more noise; accuracy in this case is increased because negative prediction is high. On the other hand, dependency matching extracts more relevant instances, increasing precision but decreasing negative prediction values.

It is important to notice that even with the usage of confidence thresholds, it is not possible to eliminate classification errors. As a result, if error propagation in the bootstrapped process is to be completely avoided, a manual checking step must be included. Although this increases resource consumption, the required effort for this manual checking step is considerably less than that of full corpus annotation. Furthermore, error and bias are expected to decrease considerably, and so are requirements for annotators' linguistic backgrounds.

2.3.3.3 Evaluation of Analysis using Expanded Training Data

For the evaluation of the expanded semantic analysis task, the improvement caused by set expansion of the original classification task proposed by Yan et al. [2008] is measured.

For the expanded training dataset obtained after each iteration, SVM classification using feature vectors is conducted, and from the output of this classification, macro-average precision, recall and F-value metrics are calculated.

The overall performance per iteration of the bootstrapped process is given in table 2.12. By analyzing the results, it is observed that a larger training dataset provides better macro-average precision, recall and F-value as expected, since the unknown feature problem is properly addressed and the existing features are recombined. However, the set size and F-value ratio is not linear, mainly because as the dataset increases in size, newly added features already exist in the dataset.

TABLE 2.12: Macro-average performance for analysis using expanded datasets

Iteration	Set Size	MA-P(%)	MA-R(%)	MA-F(%)
Initial	209	29.15	27.19	28.14
#01	969	36.74	34.95	35.82
#02	2357	42.30	35.55	38.64

The results based on micro-average precision, recall and F-value metrics are also presented in table 2.13 for comparison purposes. It is noticeable that results are fairly lower than those observed in high-frequency settings (table 2.6), which is most likely caused by the difference in dataset, the amount of training data and the richer feature set. For example, in the high-frequency setting, a ten-fold cross validation with over 12,277 relation instances was performed, and word behavior features containing frame information were used.

TABLE 2.13: Micro-average performance for analysis using expanded datasets

Iteration	Set Size	P(%)	R(%)	F(%)
Initial	209	51.80	40.36	45.37
#01	969	62.04	51.35	56.19
#02	2357	62.30	59.05	60.63

2.3.4 Manual Annotation

In the context of the proposed set expansion, data annotation consists of a simple binary classification of candidate relations into positive or negative, with the remaining steps done automatically.

The original annotation cost $\mathcal{C}$ in equations (2.1) and 2.2 is thus decreased as follows with this approach. The term T_S is 0 because entity pairs for all relations are automatically found, and T_C decreases from $\mathcal{O}(|C|)$ to a binary selection $\mathcal{O}(2)$. The equation thus

becomes

$$\mathcal{C}(L) = \sum_{L_i}^{L} \sum_{C_{k^*}}^{C^*} T_C \cdot |R_{k^*}| \cdot A(L_i) \tag{2.11}$$

where C^* is the set of infrequent relation classes, and $|R_{k^*}|$ is the set of relations for a given infrequent class C_{k^*}. This equation does not take into consideration the cost of construction for the initial set R_0, which should be small.

The main difference from equations (2.1) and 2.2 to equation (2.11) is not a decrease in cost itself, but instead the better usage of resources. Two factors are then analyzed: annotation time and number of annotated instances. For annotation time, complexity of T_C is definitely smaller for the set expansion, as $\mathcal{O}(2) < \mathcal{O}(|C|)$. However, for the number of annotated instances, it may not be guaranteed that $\sum_{C_{k^*}}^{C^*} |R_{k^*}|$ is smaller than $\sum_{C_k}^{C} |R_k|$, since the number of candidate instances for set expansion depends on the query and on filtering, and is difficult to estimate.

As for the cost versus quality trade-off, while the cost for obtaining examples of a infrequent relation class is smaller, the quality is determined by two main factors: (1) the completeness of the initial seed in terms of possible syntactic structures; and (2) the ability of the web query to return examples of the relation class given the initial set, as this should not be assumed true to all cases. The correctness of annotation is assumed in this case because of the manual annotation step.

2.4 Discussions and Conclusion

This chapter investigated bootstrapped set expansion applied for contextual semantic relations. It proposed an iterative approach for increasing the initially small set of training data by querying an unannotated corpus, matching syntactic patterns, and performing automatic and manual filtering steps. For automatic filtering, feature distance metrics were used instead of feature vectors, in order to address the sparsity of the incomplete feature space at the expense of a loss of expressibility. The usage of a multiview distance matrix was also studied, providing a unified distance value for metrics of different nature to be used for distance-based classification of new instances.

The advantadges of the proposed alternative method for reduced annotation costs is that it is not necessary to perform full corpus annotation. Given that class imbalance may result in few instances even for large annotated corpora, this research enables expansion on hand-made examples. In addition, because it is a semi-supervised approach searching for new instances in a naturally-occurring corpus, less bias is expected. The main disadvantadge is that it is not possible to find new syntactic patterns, as they

are used for identifying candidate relations from web query results. It is nevertheless possible to add examples with new patterns to the seed dataset and expand on them.

On annotation cost, the complexity was reduced from a multiclass $\mathcal{O}(|C|)$ to a binary $\mathcal{O}(2)$ with set expansion. In addition, the annotation is performed only over filtered candidate relations instead of full corpus. It is not necessarily true that the number of filtered relations is smaller than the number of relations in a corpus, but since efforts are directed towards specific classes, the usage of resources is more efficient. With the same annotation effort, it is possible to expand only sets of infrequent relation classes.

Finally, some possible improvement points for the method are discussed. First, multiview distance matrix using a linear kernel was used in this proof of concept; however, more sophisticated techniques may lead to better automatic filtering. In addition, the number of candidates per iteration was also large. It is concluded that a smaller number of instances per iteration combined with a higher number of iterations is more desirable, since it allows better control of expansion, aside from guaranteeing higher recombination of features. This might be achieved by better selecting candidates using approaches such as linguistic feature uniqueness or even randomization.

Chapter 3

Modality

3.1 Knowledge Representation

Modality is a category of linguistic meaning which is used by the speaker to express attitude and opinion towards a proposition. It provides an extra-propositional semantic component that displaces the proposition beyond the de facto world, enabling natural languages to possess expressive power beyond the actual here and now [McShane et al., 2004; Von Fintel, 2006; Nirenburg et al., 2008].

For example, consider the base proposition "*Sandy goes home at 5 p.m.*" and the modalized expressions below. In the provided sentences, the modal verb "*must*" is used for introducing obligation to the original proposition in sentence 3.1b, whereas "*might*" is used for speculation in 3.1c, and "*wants to*" for volition in 3.1d.

(3.1)	a.	Sandy goes home at 5 p.m.	Base proposition
	b.	Sandy *must* go home at 5 p.m.	Obligation = 1
	c.	Sandy *might* go home at 5 p.m.	Speculation = 1
	d.	Sandy *wants to* go home at 5 p.m.	Volition = 1

3.1.1 Modalization Mechanism

One of the ways to describe the modalization mechanism is in terms of four components, namely cue expressions, experiencer, scope and a modal value [Nirenburg et al., 2008].

Cue expressions are words that trigger the modalization mechanism. They are often expressed by modal and semimodal verbs such as "*must*" and "*have to*", as well as by

grammatical mood such as the subjunctive. As a result, modality is frequently linked to modal and mood systems in natural languages, although other expressions can also be employed as cues, with some common examples being verb phrases (ex.: "*want*"), adverbial phrases (ex.: "*perhaps*"), noun phrases (ex.: "*possibility*"), adjective phrases (ex.: "*possible*") and conjunctions (ex.: "*if*").

Cue expressions are assigned to an *experiencer*, also known as *cognizer* or *source*, which is the entity that expresses attitude towards the proposition. There are cases that more than one experiencer is observed. As a result, the notion of *private states* is introduced. A private state is an entity's state that is not open to objective verification [Wiebe et al., 2005]. For example, observe sentence 3.2 below:

(3.2) "The U.S. *fears* a spill-over," said Xirao-Nima.

In this sentence, "*fears*" expresses an opinion towards "*spill-over*" by the experiencer "*U.S.*". In turn, the clause "*The U.S. fears a spill-over*" is a state of "*Xirao-Nima*", and the whole sentence is a state of the writer. This results in a nested structure, as the original proposition is filtered by each experiencer's state.

In addition to the experiencer, cues are also said to have a modalization *scope*, which indicates the predicate that is affected by the cue expression. In the sentence 3.1b, the scope of the cue expression "*must*" is the event "*go home*"; alternatively, it can be thought of as the whole sentence. In fact, many applications use a sentence-level approach to scope because of its simplicity.

Finally, a scalar *modal value* in the interval $[0, 1]$ is also used in order to quantify the evaluation. It assumes the default value of 0 in the absence of modalization, and values closer to 1 in its presence. The modal value may be an integer or a real number, depending on the defined schema.

3.1.2 Modality Types

Modality is a quantifiable scalar displacement of a base proposition away from the actual world. Consequently, it can be understood as the union of various unidimensional values on semantic properties. Each of these dimensions is called a *modality type*.

There has been various studies concerning which types of modality exist and how they are expressed in natural languages in recent years. Some of the core cases are epistemic, evidential, deontic and dynamic modalities, as proposed by Palmer [2001].

In *epistemic modality*, speakers express judgement about the factual status of the proposition. The judgement may be of speculative, deductive or assumptive nature. In speculation or uncertainty, one of possible conclusions is expressed; in deduction, there is only possible conclusion given a knowledge base; and in assumption, a reasonable conclusion is expressed. Examples are as follows:

(3.3)	a.	He *may* be there.	Speculation = 1
	b.	He *must* be there, I've just seen him.	Deduction = 1
	c.	He *will* be there, as he always is.	Assumption = 1

In *evidential modality*, speakers indicate reported or sensorial evidence they have about the factual status of the proposition. Some examples are then given:

(3.4)	a.	I *heard* (from a news source) it'll rain.	Reported = 1
	b.	It *seems* (after seeing the sky) it'll rain.	Sensory = 1

In *deontic modality*, it is indicated how the world "ought to be", given a body of laws, set of moral principles or the like. It refers to permissive, obligative and comissive (promise or threat) meanings, and is often linked to imperative and jussive moods. For example:

(3.5)	a.	John *may* come in now.	Permissive = 1
	b.	John *must* come in now.	Obligative = 1
	c.	John *promises* to come back.	Comissive = 1
	d.	*Come* here.	Imperative = 1
	e.	*Let* me talk.	Jussive = 1

In *dynamic modality*, ability and willingness is concerned. The latter is also referred to as volition. For example:

(3.6)	a.	John *can* speak French.	Abilitive = 1
	b.	John *wants* to go.	Volitive = 1

Other common modality types in literature is *factuality* [Saurí & Pustejovsky, 2009; Baker et al., 2010]. It concerns the factual nature of an event in texts, i.e. whether the proposition has or has not happened. Factuality degress are distinguished according to certainty (certain, probable, possible, etc.) and polarity (positive or negative); it is also closely related to evidentiality. For example, in sentence 3.7, the proposition "*said*" provides hearsay evidence for "*was infatuated*".

(3.7) Newspaper reports have said that Amir was infatuated with Har-Shefi...

A final modality type is realis/irrealis. *Realis* portrays actualized situations knowable through direct perception, being usually associated with indicative moods, whereas *irrealis* portrays situations as purely in the realm of thought, being usually associated with subjunctive moods. The differentiation between the two of them is made through assertion, in which a proposition is said to not to be asserted (i.e. irrealis) if one of the following is true [Lunn, 1995]:

- The speaker has doubts about the proposition's veracity
- The proposition is unrealized
- The proposition is pressuposed

For example, in the sentence 3.8 below, it is said that the membership relation *terrorists* between *Al-Qaeda* is asserted (realis), whereas the location relation *terrorists* between *Baghdad* is not (irrealis).

(3.8) We are afraid Al-Qaeda terrorists will be in Baghdad.

3.1.3 Manual Annotation

There are several possible annotation schemas for modality, given that experiencer may or may not be annotated, scope may be annotated in different granularities, and modal values may be represented by integer or real numbers.

For the case of scope at word-level granularity and modal values represented by real numbers, the annotation task consists of: (i) identifying the set of cue expressions W_C within the text; (ii) determination of the scope of each cue expression; and (iii) determination of the modal value for each cue expression. The determination of experiencers is not considered herein.

For (i) identifying the set of cue expressions W_C within the text, all words in the set of words W need to be classified as cues or not cues, which is possible under complexity of $\mathcal{O}(2)$ per word or expression.

For (ii) determination of the scope of each cue expression, the syntax tree needs to be analyzed in order to determine the reach of modalization within the tree in terms of depth. In other words, it identifies the branch of tree for which modalization is valid. This results in a complexity of $\mathcal{O}(log|W|)$ for each cue expression within W_C.

For (iii) determination of the modal value for each cue expression, a multiclass classification is required. Given D as the number of decimal places (precision) of the modal value, the complexity is given by $\mathcal{O}(10^D)$ for each cue expression within W_C.

Given that annotation has to be done for each modality type M_j within the set of modality types M, the total annotation cost $\mathcal{C}$ for this task for a set of languages L is based on equation (1.1) and may be roughly estimated by

$$\mathcal{C}(L) = \sum_{L_i}^{L} \sum_{M_j}^{M} (T_{cue} \cdot |W| + T_{scope} \cdot |W_C| + T_{modal} \cdot |W_C|) \cdot A(L_i, M_j) \tag{3.1}$$

where T_{cue} is the averate time to manually determine one cue expression (complexity of $\mathcal{O}(2)$), T_{scope} is the averate time to manually identify the scope for one cue expression (complexity of $\mathcal{O}(log|W|)$), and T_{modal} is the average time to manually determine the modal value for one cue expression (complexity of $\mathcal{O}(10^D)$).

Depending on the application, simplified annotation schemas are also possible. Two simplifications are then presented.

First is the simplification of modal values from real numbers $[0, 1]$ to integers $\{0, 1\}$. This decreases the complexity of T_{modal} from $\mathcal{O}(10^D)$ to 0, because every scope is then assumed to have modal value equal to 1.

Second is the simplification of scope from word-level to sentence-level. This decreases the complexity of T_{scope} from $\mathcal{O}(log|W|)$ to 0, because the existence of one cue expression forces the sentence's modalization. Hence, determination of modalized sentences can be done automatically at no cost in this case.

With these two simplifications, the equation 3.1 can be reduced to the equation

$$\mathcal{C}(L) = \sum_{L_i}^{L} \sum_{M_j}^{M} T_{cue} \cdot |W| \cdot A(L_i, M_j) \tag{3.2}$$

3.2 Semantic Analysis

3.2.1 Problem Statement

In state-of-the-art works in modality, semantic analysis is concerned with identyifying modalized text spans at word-level or sentence-level scopes. This is normally achieved by one of the following two approaches:

- *One-step approach*: Classifies scopes as modalized or not, using words within the boundaries of a *fixed scope* (ex.: sentence-level) as classifier features
- *Two-step approach*: First classifies expressions as cues or non-cues using linguistic features; then uses cue expressions to evaluate modalization of scopes (if at least one cue expression exists within the scope, it is considered modalized)

In either approach, some extent of word disambiguation is crucial. In fact, this has been identified as one of the greatest challenges in modality [Von Fintel, 2006]. In an ideal setting, the identification of modalization at the sentence-level is trivial because of one-to-one correlation with cue expressions; however, this is not the case in a real-world setting, since ambiguity may cause non-modalized sentences to be incorrectly evaluated. In the example below, the same cue expression "*must*" may assume a meaning of speculation as in sentence 3.9a, or a meaning of obligation as in sentence 3.9b.

(3.9) a. Sandy *must* be at home. Speculation = 1

b. Sandy *must* go home. Obligation = 1

Current WSD techniques are designed to deal with nouns, verbs, adjectives and adverbs, which are solved using local context. For example, if the noun "*bank*" is used in the sense of "*financial institution*", it is usually surrounded by words such as "*credit*" and "*money*"; if it is used in the sense of "*land alongside a river*", then it is surrounded by words such as "*river*" and "*water*".

However, in examples 3.9a and 3.9b, it is clear that standard WSD techniques based on local context may not be applied in the case of modal disambiguation. There is an information gap between syntactic and semantic level which is not addressed by available morphological, syntactic, lexical, ontological and semantic linguistic features. As a result, given the lack of deep semantic or pragmatic features, disambiguation of cue expressions is performed on a best-effort basis

3.2.2 Corpora and Systems

Uncertainty is a modality type that has received considerable attention from the NLP community. Tasks concerning the extraction of uncertainty semantics consist of identifying the scope of modalization and its respective modal value in raw text.

Early work by Light et al. [2004] defined a hand-crafted list of cue expressions, while Medlock & Briscoe [2007] and Szarvas [2008] worked on semi-automatical classification of sentence modal value.

The CoNLL-2010 Shared Task [Farkas et al., 2010] defined two uncertainty corpora, namely the Biological corpus and the Wikipedia corpus, for which participating systems proposed solutions for the uncertainty extraction task.

The Biological corpus consists of annotating *hedging* in scholar articles from the FlyBase corpus [Medlock & Briscoe, 2007], BioScope [Vincze et al., 2008], BMC Bioinformatics and PubMedCentral. Hedge cue expressions are words with speculative content, such as auxiliary verbs ("*may*", "*might*", etc.), speculative verbs ("*suggest*", "*indicate*", etc.), adjectives and adverbs ("*probable*", "*likely*", etc.), conjunctions ("*or*", "*either*", etc.), among others. An example is given:

(3.10) Mild bladder wall thickening *raises the question of* cystitis.

The Wikipedia corpus consists of annotating *weasels* in articles from the online encyclopedia. Weasels are words that vaguely specify the source of information, which represent author opinion or unreliable information source. Typical examples are expressions denoting uncertainty ("*probable*", "*likely*", etc.), generalization ("*widely*", "*traditionally*", etc.), obviousness ("*clearly*", "*arguably*", etc.), passive forms with dummy subjects ("*it is claimed*", etc.), numerically vague expressions ("*certain*", "*numerous*", etc.), among others. An example is given:

(3.11) *Some people* claim that this results in a better taste than that of other diet colas.

The CoNLL-2010 ST proposes two tasks: task 1, which defines scope at sentence-level, providing data manually annotated with cue expressions and sentence binary modal value; and task 2, which defines scope at word-level, providing data manually annotated with cue expressions, modalization scope and scope binary modal value. Task 1 is the focus of this work and will be the only task considered hereon.

Training and testing data are provided for each corpus. Table 3.1 presents a summary of the two corpora for task 1, stating the number of sentences, percentage of uncertain sentences, number of distinct cue expressions and number of ambiguous cue expressions, i.e. words whose raw form may or may not be cues [Georgescul, 2010].

Evaluation of a system in the task is based on precision, recall and F-value of sentence binary modal value. A true positive is when the modal value of 1 is obtained when the modal value 1 is expected for a given sentence (i.e. at least one cue expression was found within an uncertain sentence), a false positive is when 1 is obtained when 0 is expected (i.e. at least one cue expression was found within a certain sentence), and so on. As a

TABLE 3.1: CoNLL-2010 ST corpora

Corpus	# Sentences	% Uncertain	# Distinct Cues	# Ambiguous Cues
Biological training	14541	18%	168	96
Biological testing	5003	16%	—	
Wikipedia training	11111	22%	1912	188
Wikipedia testing	9634	23%	—	

result, precision measures the number of correctly identified modalized sentences within all sentences that were identified as modalized, whereas recall measures the number of correctly identified modalized sentences within all sentences that should have been identified as modalized.

A naïve baseline system [Georgescul, 2010] is given in table 3.2. It classifies any sentence containing at least one cue expression found in training data without performing any disambiguation. As expected, recall values are high and precision values are low, because of the high number of false positives and low number of false negatives due to ambiguity. In this benchmark, it makes sense to measure precision, recall and F-value in training data in order to account for the importance of disambiguation.

TABLE 3.2: CoNLL-2010 ST naïve baseline

Corpus	Precision	Recall	F-value
Biological training	46%	99%	63%
Biological testing	42%	98%	59%
Wikipedia training	32%	96%	48%
Wikipedia testing	45%	86%	59%

A summary of the participating systems of the shared task, together with their methods, approaches and linguistic features, are described in table 3.3 [Farkas et al., 2010].

Other notable works on modality are as follows. OntoSem [Nirenburg et al., 2008] is a semantic analyzer which uses polarity, volition, obligation, belief, potential, permission and evaluative modality as part of its semantic module. The MPQA Opinion Corpus [Wiebe et al., 2005] is a corpus used for opinion mining, annotated with private states at word- and phrase-levels and defining two types of frames which distinguish opinion-oriented and factual material. The FactBank [Saurí & Pustejovsky, 2009] is a corpus

TABLE 3.3: CoNLL-2010 ST Task 1 participating systems

Name	Method	Appr	Features					
			dic	ort	lem	pos	chk	dep
Clausen	ME	BoW	—	—	x	x	—	—
Chen	ME	BoW	x	—	x	x	—	—
Fernandes	ETL	SL	—	—	x	x	x	—
Georgescul	SVM	BoW	x	—	—	—	—	—
Ji	AP	TC	—	—	—	x	—	—
Kilicoglu	Manual	TC	x	—	x	—	—	x
Li	CRF	SL	—	—	x	x	x	—
Morante	SVM/kNN	TC/SL	x	—	x	x	x	x
Prabhakaran	CRF	SL	x	—	x	x	—	x
Rei	CRF	SL	x	—	x	x	—	x
Sáanchez	SVM	BoW	x	—	x	x	—	x
Shimizu	BPM	SL	—	x	x	x	x	—
Szidarovsky	CRF	SL	x	x	x	—	—	—
Täckström	SVM	BoW	—	—	x	x	x	x
Tang	CRF/SVM	SL	x	x	x	x	x	—
Tjong	NB	TC	—	—	—	—	—	—
Velldal	ME	TC	x	—	x	x	—	x
Vlachos	LogReg	TC	x	—	x	x	—	x
Zhang	CRF	SL	x	—	x	x	x	x
Zhao	CRF	SL	x	—	x	x	x	—
Zheng	CRF/ME	SL	—	—	x	x	x	—
Zhou	CRF	SL	x	—	x	x	x	—

Approaches are bag-of-words model (BoW), sequence labeling (SL) and token classification (TC). Methods are Maximum Entropy (ME), Entropy Guided Transformation Learning (ETL), Support Vector Machines (SVM), Average Perceptron (AP), kNN (k-Nearest Neighbors), Conditional Random Fields (CRF), Bayes Point Machines (BPM), Naïve Bayes (NB) and Logistic Regression (LogReg). Linguistic features are external dictionaries (dic), orthographical information (ort), stem or lemma (lem), part-of-speech tag (pos), syntactic chunk (chk) and constituent or dependency parse (parse).

annotated with factuality information, being used as a second layer on top of TimeBank [Pustejovsky et al., 2003b]. Finally, the BCCWJ Modality Corpus [Matsuyoshi et al., 2010] consists of modality tags built upon the Balanced Corpus of Contemporary Written Japanese (BCCWJ) [Maekawa, 2007].

3.2.3 Experiments and Results

The systems for task 1 of the CoNLL-2010 ST stated in table 3.3 attempted to classify sentences as modalized or not for the Biological and Wikipedia corpora. Classification results in terms of precision, recall and F-value were obtained as indicated in table 3.4. The last column, F_{avg}, is an arithmetic average between the F-values for the two corpora. The best three systems for each corpus are highlighted.

TABLE 3.4: CoNLL-2010 ST Task 1 results

Name	Biological			Wikipedia			F_{avg}(%)
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	
Clausen	79.33	80.63	79.97	75.12	42.03	53.90	66.94
Chen	74.85	79.11	76.92	**67.97**	**49.69**	**57.41**	67.17
Fernandes	70.07	71.14	70.60	—	—	—	—
Georgescul	69.07	91.01	78.54	**72.04**	**51.66**	**60.17**	69.36
Ji	79.45	76.33	77.86	**62.66**	**55.28**	**58.74**	68.30
Kilicoglu	92.07	74.94	82.62	67.90	46.02	54.86	68.74
Li	**90.40**	**81.01**	**85.45**	88.35	31.92	46.89	66.17
Morante	80.54	83.29	81.89	80.55	44.49	57.32	69.61
Prabhakaran	67.54	19.49	30.26	87.95	28.42	42.96	36.61
Rei	83.75	84.18	83.96	—	—	—	—
Sáanchez	—	—	—	68.28	46.24	55.14	—
Shimizu	88.08	82.28	85.08	—	—	—	—
Szidarovsky	70.28	91.01	79.32	—	—	—	—
Täckström	87.05	83.42	85.20	78.31	42.84	55.38	**70.29**
Tang	**85.03**	**87.72**	**86.36**	82.28	41.36	55.05	**70.71**
Tjong	74.30	87.09	80.19	74.02	42.97	54.38	67.29
Velldal	85.48	84.94	85.21	—	—	—	—
Vlachos	85.48	84.94	85.21	—	—	—	—
Zhang	82.59	84.68	83.63	76.58	44.36	56.18	**69.91**
Zhao	83.44	84.81	84.12	—	—	—	—
Zheng	73.31	90.76	81.11	76.27	43.60	55.48	68.30
Zhou	**86.49**	**85.06**	**85.77**	85.27	36.53	51.14	68.46

3.3 Alternative Semantic Analysis

3.3.1 Problem Statement

3.3.1.1 Three-Step Approach to Modality Analysis

Two points of improvement in the current approach are discussed hereon. Given the correlation between cue expressions and modalized sentences, the first point is to determine cues based solely on sentence modal value. In other words, it should be possible to perform the same task without using cue word annotation at all. Such approach is interesting for the modality domain, since it decreases the annotation effort for each one of the various different modality types indicated in section 3.1.2.

The second point regards unreliable cue expression annotation from a machine learner point of view. It has been observed in the Wikipedia corpus that 63% of cue types presented unique occurrences [Morante et al., 2010]. This translates into sparsity and feature inconsistency in the one-step approach, and into unreliable examples for the classifier in the two-step approach. As a result, aside from Georgescul [2010], all systems in the CoNLL-2010 ST were outperformed by the naïve baseline in table 3.2 for

the Wikipedia corpus. Guaranteeing corpus reliability is difficult, since it can only be assessed after a considerable amount of effort has been put in annotation.

It is also argued herein that cue expression annotation is language-dependent semantics, and not interlingual. The exact linguistic element to which modalization is attached greatly differs from language to language. In the example below, volition is attached to the verb "*want*" in the English sentence 3.12a, whereas it is attached to the verb conjugation in the Japanese sentence 3.12b.

(3.12) a. I *want* to go there.

b. Asoko ni iki*tai*.

One solution to the presented two points is *cue selection*, which is the process of automatically choosing which expressions from the corpus are actually cues. While obviously addressing the first point of improvement, it discards noise and cue expressions that are potentially harmful for a machine learner for the second point.

Hence, this becomes a *three-step approach*:

1. Word sense disambiguation on training and testing data
2. Cue expression selection from training data
3. Evaluation of sentence modal value in testing data based on cue expressions selected from training data

Unlike the two-step approach, disambiguation and selection are decoupled from each other (the last step is the same). The adoption of a separate cue selection is interesting for a variety of reasons. First, modality analysis becomes independent from cue annotation, which is not reliable and not interlingual. Second, unlike the one-step approach, the cue expression is identified. Third, unlike the two-step approach using supervised classification, this work relies on the correlation between cues and sentence modalization, which is the best indicator of cue existence, rather than features which are not good descriptors of the linguistic phenomenon. Aside from these points, other advantadges are:

- It is possible to analyze intermediate results (i.e. how cue disambiguation performs and how it affects cue selection)
- Cue selection acts as a layer of correction of disambiguation errors

- Cue expression disambiguation research can be conducted separately
- It is later shown that this approach requires less linguistic features

When focusing on cue selection, this work aims to decrease annotation costs by requiring annotation at the sentence-level instead of word-level. This results in a smaller number of instances to be annotated. As for other strategies to decreasing costs, active learning may not be used under this domain. This is because any method based on sampling risks disregarding a sentence with an unknown cue expression. Medlock & Briscoe [2007] and Szarvas [2008] proposed a semi-automatic approach to cue expression annotation. However, given that sentence alignment techniques for parallel corpora [Moore, 2002] exist and are widely employed in SMT, it is possible to propagate sentence-level modalization annotation when using a cue selection-based approach.

3.3.1.2 Formalization of the Cue Selection Problem

The problem of cue selection is then formalized as follows. Let $\mathcal{D}$ be the set of uncontextualized disambiguated expressions (dictionary). Each sentence s_j in training data is modeled as a bag-of-words $s_j \subset \mathcal{D}$ and has a modal value $mod(s_j) \in \{0, 1\}$. Let $\mathcal{S} \subset \mathcal{D}$ then be the set of selected cue expressions. If a word $w \in s_j$ is selected ($w \in \mathcal{S}$), then $mod(s_j) = 1$; otherwise, $mod(s_j) = 0$.

Cue selection is then the decision problem of choosing a set $\mathcal{S}$ which optimizes precision, recall and F-value of the binary classification on $mod(s_j)$ over all training sentences. As a decision problem, its computational complexity is expected to be $\mathcal{O}(2^m n)$, where m is the number of contextualized expressions in the corpus and n is the number of sentences.

3.3.1.3 Difficulties of the Cue Selection Problem

Three difficulties of cue selection are explained. They are: noise, cue-bounded modalization and selection equivalence.

First and foremost, the concepts of precision (P), recall (R) and F-value (F_1) of a single expression $w \in \mathcal{D}$ are defined. They are the P, R and F_1 calculated over the sentences which contain a word w in case only w is selected ($\mathcal{S} = \{w\}$).

Noise then regards the difficulty of correctly discriminating cues from non-cues. Expressions can be classified into the following seven cases according to frequency and ambiguity (examples are given for the Biological uncertainty corpus):

(A) Frequent unambiguous cues: High P, medium R (ex.: "*suggest*")

(B) Frequent ambiguous cues: Low P, medium R (ex.: "*consider*")

(C) Infrequent unambiguous cues: High P, low R (ex.: "*speculate*")

(D) Infrequent ambiguous cues: Low P, low R (ex.: "*assume*")

(E) Frequent non-cues co-occurring with cues: Medium P, high R (ex.: "*that*")

(F) Infrequent non-cues co-occurring with cues: Variable P, low R

(G) Non-cues not co-occurring with cues: Zero P, zero R

From these categories, it is understood that noise may present higher F_1 than some cue expressions. In fact, it is difficult to differentiate case (E) from case (B), as well as case (F) from both (C) and (D) when selecting cue expressions.

Cue-bounded modalization is a difficulty defined as the fact that if an expression from a given sentence s_j is already selected, any other selection will not further affect the modality status of s_j. Because of this property, if non-cues are not properly filtered, the number of false positives will increase dramatically. Additionally, considering a word w and a set of expressions $W = \{w1, w2, \dots, \}$, if the selection of w has higher F_1 than each $\{w1, w2, \dots, \}$ individually, but lower than W as a whole (true when false positives for $w1, w2, \dots$ are overlapping), it is not theoretically possible to determine the global optimum in a timely manner.

Finally, *selection equivalence* is defined as follows. Given two different sets of expressions W_1 and W_2 which pertain to the same sentences, these sets obtain exactly the same selection and consequently same F_1. Standard selection alone is not able to address this equivalence, since the choice between W_1 and W_2 has to be based on empirical observations on data.

3.3.2 Cue Selection Optimization

3.3.2.1 Overview

Cue selection is designed as a process of iterative selections of expressions. This work uses an optimization-based approach, since the best available indicator of cue existence is the correlation between cue expression and sentence-level modality value, rather than linguistic features, given that features that properly disambiguate and select cue expressions are not currently available. By optimizing, selection is directly maximized for

information extraction metrics, whereas in supervised classification, selection is modeled in terms of such features.

Each iteration of the selection process is divided into searching a candidate cue expressions based on some *heuristics*, and selecting/deselecting expressions based on improvement of the global *objective function*. The process is halted when there are no expressions whose selection/deselection can further increase the objective function for the current iteration.

Heuristics are used for approximating the choice of the best selected cue expression, addressing the combinatorial search problem. The objective function serves as the value to be optimized by the iterative process. For both cases, the F_1 metric seems to be the most obvious choice.

However, for heuristics, F_1 for case (E) is higher than cases (A) and (B), causing noise to be selected earlier. Moreover, because of selection equivalence, the algorithm may choose a set of expressions which leads to worse results in sentence modalization, despite having defined the deselection operation. A better approach is then to guarantee as many correct selections as possible in early steps by using F_β as primary heuristic ($F_\beta = 0.25$ determined arbitrarily) and F_1 as fallback. With a larger initial weight to precision, frequent cues have higher confidence and are thus selected earlier. This also guarantees faster convergence. An algorithm is detailed in a following section.

It should be observed that cue selection (CS) is similar to feature selection (FS) methods [Dash & Liu, 1997; Yu & Liu, 2003; Fleuret, 2004], especially heuristic wrappers. However, the two domains differ considerably. While FS is concerned with feature redundancy, optionally removing the less informative of correlated feature sets, CS is concerned with the task-specific selection equivalence, in which it is crucial to remove all non-cue expressions because of cue-bounded modalization. In addition, the only way to filter non-cues is to identify the correct cue within the same sentence, favoring a confidence-based approach over a strictly correlation-based one. Heuristic CS is also similar to decision list learning [Chakravarthy et al., 2008; Goodman, 2002], which determines the order that rules of a rule-based system are applied. The main difference between the two is that heuristic CS is concerned not only with the order of selection, but also determining which expressions are not selected.

3.3.2.2 Algorithm

Let each expression in the dictionary $w_i \in \mathcal{D}$ be described by a sparse binary vector $\vec{u}_i$, where u_{ij} indicates whether w_i exists in sentence s_j. Let $\mathcal{S}$ be the set of selected

expressions (current state of selection), and $\mathcal{S}'$ be the set of expressions that are not currently selected ($\mathcal{S}' = \mathcal{D} - \mathcal{S}$). Let also Γ be a function that given a set of expressions W, returns a binary vector as follows:

$$\Gamma(W) = \vec{u}_1 \vee \cdots \vee \vec{u}_{|W|} \tag{3.3}$$

The vector $\vec{u} = \Gamma(\mathcal{S})$ is then defined as a sparse binary vector whose elements indicate the sentences that are selected given current state $\mathcal{S}$.

Let $\vec{u}^+$ be a vector indicating sentences s_j for which $mod(s_j) = 1$ in training data, let $\neg\vec{v}$ be the vector whose binary elements are equal to the negation of those in $\vec{v}$, and let $|\vec{v}|$ be a function that counts the amount of 1's in a binary vector $\vec{v}$. For any given vector $\vec{v}$, it is possible to calculate precision, recall and F_β measure, as indicated below:

$$tp(\vec{v}) = |\vec{u}^+ \wedge \vec{v}| \tag{3.4}$$

$$fp(\vec{v}) = |\neg\vec{u}^+ \wedge \vec{v}| \tag{3.5}$$

$$fn(\vec{v}) = |\vec{u}^+ \wedge \neg\vec{v}| \tag{3.6}$$

$$P(\vec{v}) = \frac{tp(\vec{v})}{tp(\vec{v}) + fp(\vec{v})} \tag{3.7}$$

$$R(\vec{v}) = \frac{tp(\vec{v})}{tp(\vec{v}) + fn(\vec{v}))} \tag{3.8}$$

$$F_\beta(\vec{v}) = \frac{(1+\beta^2) \cdot P(\vec{v}) \cdot R(\vec{v})}{\beta^2 \cdot P(\vec{v}) + R(\vec{v})} \tag{3.9}$$

Algorithm 2 provides the proposed cue selection method. It starts with an empty selected set $\mathcal{S}$ and an unselected set $\mathcal{S}'$ containing all expressions (lines 3 to 5). In each step, vectors corresponding to next states are calculated for every expression (lines 9 to 10). These vectors are used in order to determine which expressions should be used for updating the current state.

Given candidate expressions chosen by heuristics F_β and F_1 (lines 11 to 12), one of them is selected/deselected if and only if the operation generates a gain in the global objective F_1 (lines 13 to 15). As a result, the algorithm selects the candidate expression if it improves F_1, deselects when a previous selection was harmful to F_1, and stops when no operation can improve F_1, returning the current state (line 17). Input expressions are filtered in the beginning (line 2). Expressions whose number of true positives are smaller than a threshold θ are excluded from the input, since the ratio of noise to cue

Algorithm 2 Cue selection

1: **function** SELECTCUES($W = \{w_i\}$, $\{\vec{u}_i\}$, $\vec{u}^+$)
2: $Filter(W)$
3: $\vec{u} \leftarrow (0, 0, \ldots, 0)$
4: $\mathcal{S} \leftarrow \{\}$
5: $\mathcal{S}' \leftarrow W$
6: **while** *true* **do**
7: $i_\beta \leftarrow 1$, $i_1 \leftarrow 1$
8: **for** $i \leftarrow 2$ **to** m **do**
9: **if** $w_i \in \mathcal{S}$ **then** $\vec{u}'_i \leftarrow \Gamma(\mathcal{S} - \{w_i\})$
10: **else** $\vec{u}'_i \leftarrow \Gamma(\mathcal{S} + \{w_i\})$
11: **if** $F_\beta(\vec{u}'_i) > F_\beta(\vec{u}'_{i_\beta})$ **then** $i_\beta \leftarrow i$
12: **if** $F_1(\vec{u}'_i) > F_1(\vec{u}'_{i_1})$ **then** $i_1 \leftarrow i$
13: **if** $F_1(\vec{u}_{i_\beta}) > F_1(\vec{u})$ **then**
14: **if** $w_{i_\beta} \notin \mathcal{S}$ **then** $Select(w_{i_\beta})$ **else** $Unselect(w_{i_\beta})$
15: **else if** $F_1(\vec{u}_{i_1}) > F_1(\vec{u})$ **then**
16: **if** $w_{i_1} \notin \mathcal{S}$ **then** $Select(w_{i_1})$ **else** $Unselect(w_{i_1})$
17: **else return** $\mathcal{S}$
18: $\vec{u} \leftarrow \Gamma(\mathcal{S})$

expressions occurrence is too large for expressions with small number of true positives. The value θ is obtained empirically from the training data.

Some properties obtained from such approach is as follows. First, the proposed selection does not require cue annotation and is completely language independent; it is also flexible as to what can be encoded as input expressions. Consequently, it is independent of NLP tools and works with sentence splitter, tokenizer and stemmer at the very minimum.

In addition, the time complexity of the algorithm is given by $\mathcal{O}(kmn)$, where k is the number of selected expressions $|\mathcal{S}|$ plus twice the number of deselection operations. Since the ratio of cues in the vocabulary is small ($|\mathcal{S}| \ll m$) and deselections are rare, complexity is subquadratic in a real setting. The space complexity is $\mathcal{O}(mn)$, but this can be decreased with sparse vector implementations.

The algorithm also deals with overfitting by means of the confidence-based selection. If noise is to be selected, it will be selected in later stages of the process. This results in only noise with few instances being selected, which represents a small error. In addition, overfitting is also partially addressed by the filtering step (line 2).

Finally, the algorithm is also easily parallelizable, as each run of the inner loop (lines 9 to 12) is independent. Different distributed nodes can therefore calculate F_β and F_1

individually, and the controller can identify the candidate expressions for each iteration and update the state.

3.3.3 Experiments and Results

3.3.3.1 Experimental Setting

The proposed method is used for both Biological and Wikipedia corpora from CoNLL-2010 ST. Intermediate results on cue selection are qualitatively analyzed, and final results on sentence-level modalization are compared against baseline systems described in section 3.2.2.

For the three-step approach mentioned in section 3.3.1.1, raw words and expressions from training data are first disambiguated. In this work, seven different sets of linguistic information are used:

- `A`: Sentence splitter, tokenizer and stemmer (minimum requirement for processing)
- `B`: Sentence splitter, tokenizer, POS tagger and lemmatizer
- `C`: Disambiguation using WordNet::SenseRelate [Pedersen & Kolhatkar, 2009], which measures semantic relatedness between a word and its neighbors using WordNet. For every content word of a sentence, similarity and relatedness measures (ex.: WordNet path lengths augmented with information content) are calculated for all surrounding word senses, and the sense with the highest sum of scores is chosen
- `D`: Disambiguation using semantic graph connectivity over the underlying knowledge base BabelNet [Navigli & Lapata, 2010; Navigli & Ponzetto, 2012]
- `E`: Union of `B` and `D`, adding different levels of granularity, since some disambiguated word senses may be too granular for the task
- `F`: Union of `A`, `B` and `D`, adding different levels of granularity
- `G`: Subset of `F`. Words within training cue expressions are encoded using `A`, `B` and `D`; they are then combined, forming the same cue expressions as in training data, but with different encodings for each word

Unigrams, bigrams and trigrams are used in order to account for local context for each of the previous sets. This is illustrated in tables 3.5 and 3.6 for the hedge cue expression "*is believed*". Encoding systems are identified by `XN` herein, where `X` is the one-character

TABLE 3.5: Examples of disambiguated expressions for cue selection (unigrams)

Set	Unigrams	
A	is	believ
B	be/VB	believe/VB
C	be/VB/v1	believe/VB/v1
D	be/VB-bn:83181v	believe/VB-bn:83369v
E	be/VB	be/VB-bn:83181v
F	is	be/VB-bn:83181v

TABLE 3.6: Examples of disambiguated expressions for cue selection (bigrams)

Set	Bigrams	
A	is	is_believ
B	be/VB	be/VB_believe/VB
C	be/VB/v1	be/VB/v1_believe/VB/v1
D	be/VB-bn:83181v	be/VB-bn:83181v_believe/VB-bn:83369v
E	be/VB_believe/VB	be/VB-bn:83181v_believe/VB
F	is_believ	be/VB-bn:83181v_believ

code for each set and N is the number of n-grams. For example, A2 stands for bigrams using stems only.

The disambiguated expressions are then used as input of algorithm 2. It should be noted that the algorithm works with raw forms. As a result, disambiguated n-grams must be encoded into a single string. Such representation enables the usage of various different WSD methods with no effort on adapting the algorithm, although it is not possible to input more complex representations such as trees, which are not expected for modality.

The algorithm then selects which of the disambiguated expressions found in training data are cues, and use them in order to evaluate sentence modalization in testing data.

3.3.3.2 Evaluation of Cue Selection

This section analyzes the output of cue selection before using it for sentence modalization evaluation. Tables 3.7 and 3.8 provide some of the resulting selected cue expressions for different settings. Each row corresponds to one iteration of the algorithm; iteration number, executed operation (select or deselect, with "*" when fallback is used), the number of positives and negatives, and encoded expression are also indicated.

Resulting selected cue expressions can be used for better understanding the mechanism of the algorithm. By using $F_{\beta=0.25}$ as primary heuristics for selection, the priority is given to expressions with higher precision, eliminating frequent non-cues from early selections

TABLE 3.7: Examples of cue selection output using `B2` (Biological corpus)

Iter	Oper	F_1	# Positives	# Negatives	Cue
1	sel	41.3	683	1	suggest/VB
2	sel	59.3	558	0	may/MD
3	sel	68.8	315	2	indicate/VB_that/IN
4	sel	73.5	158	0	appear/VB_to/TO
5	sel	77.1	124	1	whether/IN
6	sel	79.1	106	0	might/MD
7	sel	80.2	41	0	be/VB_think/VB
8	sel	81.2	45	0	seem/VB
9	sel	82.0	36	0	be/VB_likely/JJ
10	sel	82.6	30	0	possibly/RB
23	sel	86.4	13	0	not/RB_appear/VB
25	sel*	87.1	50	17	putative/JJ
26	sel*	87.7	81	16	possible/JJ
27	sel*	88.1	81	5	likely/JJ
28	sel*	88.5	26	2	potentially/RB

TABLE 3.8: Examples of cue selection output using `B2` (Wikipedia corpus)

Iter	Oper	F_1	# Positives	# Negatives	Cue
1	sel	32.0	527	288	some/DT
2	sel	34.7	61	4	probably/RB
3	sel	48.5	414	272	many/JJ
4	sel	49.3	27	5	be/VB_believe/VB
9	sel	52.5	25	8	possibly/RB
10	sel*	54.8	150	184	several/JJ
11	sel*	56.6	157	214	most/RB
12	sel*	57.8	122	71	consider/VB
14	sel*	59.1	172	196	have/VB_be/VB
30	sel*	61.8	24	22	might/MD
31	desel*	61.9	172	196	have/VB_be/VB
32	sel*	62.0	56	21	suggest/VB
43	sel*	62.9	14	2	possible/JJ_that/IN
44	sel*	63.1	26	18	whether/IN
77	sel*	64.3	12	9	hard/JJ

and thus avoiding selection equivalence. After some selections, the fallback heuristic is then used, since the larger weight to precision generates candidates with poorer F_1.

In addition, threshold value θ can be empirically obtained from training data at this stage by testing different values and analyzing the resulting selection. It is observed that although satisfactory results were obtained by using this filter, it is not possible to completely isolate noise from cue expressions by using such simple criteria.

Table 3.9 gives information on the state of the algorithm upon halting for all feature sets used. The empirically defined θ, F_1 metric and training times are stated for each

corpus. It should be noticed that testing times are very low, and are thus omitted.

TABLE 3.9: Results of cue selection upon algorithm halt (after step two)

Name	Biological			Wikipedia		
	θ	$F(\%)$	t(s)	θ	$F(\%)$	t(s)
A1	3	87.16	4.927	11	61.27	4.261
A2	6	90.75	12.684	11	63.32	9.346
A3	6	90.78	18.283	9	64.64	19.210
B1	3	89.48	6.799	10	62.30	3.840
B2	11	90.56	5.421	10	64.33	10.791
B3	11	90.62	8.403	10	64.58	12.827
C1	3	89.42	11.586	6	64.04	14.928
C2	10	89.89	7.103	7	65.37	24.549
C3	10	89.92	10.326	5	67.21	50.077
D1	3	89.59	18.656	5	64.38	36.899
D2	5	91.21	33.934	10	63.64	17.936
D3	5	91.24	43.252	11	63.21	15.806
E1	4	89.57	17.256	9	63.31	17.770
E2	11	90.77	20.673	10	64.73	25.758
E3	7	91.36	77.813	10	65.03	49.840
F1	4	89.69	25.177	9	63.60	40.255
F2	7	91.55	84.160	16	63.09	27.706
F3	13	90.70	500.733	16	63.20	143.477
G3	4	90.31	2.199	4	63.01	14.926

Selection training times are given for an Intel i5, 2.3GHz, 4GB RAM environment.

Aside from the computationally expensive WSD and from F3, which has a large m due to naïvely combining sets of linguistic information, it is observed that processing is very fast because of the $\mathcal{O}(kmn)$ complexity. Running times are then compared with other systems. Georgescul [2010] used a Pentium 4, 3.2GHz, 3GB RAM environment and took 4 hours for parameter tuning, 19.5s for training and 2.6s for testing in the biological corpus, and 13h parameter tuning, 49.1s training and 21.5s testing in the Wikipedia corpus. Morante et al. [2010] used a MacOS X, 2.2GHz, 2GB RAM, obtaining 22.5h when training on the Wikipedia corpus, and used an Intel Xeon, 2.8GHz, 8GB RAM, obtaining 10.44s when testing on the biological corpus, in a cross domain approach. Zhao et al. [2010] used an Intel Xeon, 2.0GHz, 6GB RAM and obtained training and testing in under 8 minutes in the biological corpus.

3.3.3.3 Evaluation of Sentence-Level Modalization

Cue expressions obtained from the previous step are then used for detecting modalized sentences in testing data, with experimental results indicated in table 3.10. These detected sentences are compared against CoNLL-2010 ST baselines in table 3.4, noticing that except for `G3`, cue annotation is not used. Top three results for each dataset are highlighted.

TABLE 3.10: Results of sentence modalization using cue selection (after step three)

Name	Biological			Wikipedia			F_{avg}(%)
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	
`A1`	74.10	83.29	78.43	64.50	58.55	61.38	69.91
`A2`	87.14	81.52	84.24	63.65	57.07	60.18	72.21
`A3`	87.26	81.52	84.29	63.44	57.48	60.31	72.30
`B1`	76.00	84.56	80.05	65.99	55.24	60.14	70.10
`B2`	**88.90**	**81.14**	**84.84**	**67.97**	**56.89**	**61.94**	**73.39**
`B3`	88.89	81.01	84.77	68.43	55.51	61.30	73.04
`C1`	74.53	80.76	77.52	61.43	57.25	59.27	68.40
`C2`	87.55	76.58	81.70	63.13	57.56	60.22	70.96
`C3`	87.55	76.58	81.70	62.85	58.01	60.34	71.02
`D1`	75.09	82.78	78.75	61.49	58.46	59.94	69.35
`D2`	87.29	80.00	83.49	66.12	55.64	60.43	71.96
`D3`	87.41	80.00	83.54	66.09	54.34	59.64	71.59
`E1`	75.28	84.81	79.76	64.63	57.65	60.94	70.35
`E2`	88.50	80.89	84.52	66.84	56.76	61.39	72.96
`E3`	88.07	81.27	84.53	65.84	57.03	61.12	72.83
`F1`	75.42	85.06	79.95	63.47	58.86	61.08	70.52
`F2`	**88.13**	**81.77**	**84.83**	68.25	56.00	61.52	**73.18**
`F3`	**89.06**	**81.39**	**85.05**	**68.26**	**56.22**	**61.66**	**73.36**
`G3`	80.10	84.05	82.03	**69.14**	**58.86**	**63.59**	72.81

Comparisons can be made to CoNLL-2010 Shared Task results from table 3.4.

First and foremost, it is observed that `B2` and `F3` obtained promising results in both datasets: `B2` uses little resources (only lemmatizer and POS tagger), whereas `F3` uses WSD and combines linguistic features, which increases preprocessing and training time.

When comparing to CoNLL-2010 ST participants in table 3.4, the proposed method outperforms all systems for the Wikipedia dataset (with `B2` obtaining +1.77 percentage points when compared against the best baseline, and `G3` obtaining +3.42pp) and ranks 8th for the Biological dataset (`B2` -1.52pp, `F3` -1.31pp). It is also the best overall system, if F_1 values for the two datasets were to be averaged (`B2` +2.68pp).

In addition, bigrams and trigrams provide much better results than unigrams for the biological corpus, although such trend is not observed for the Wikipedia dataset. Specifically for the Wikipedia corpus, it is also possible to verify that this method obtains higher values for annotation which is less reliable from a machine learner point-of-view. As observed by the lower F-values of the baseline, cues are more ambiguous in this dataset, whereas features are not able to properly disambiguate expressions such as "*some*", "*probably*" and "*many*" (table 3.8). This also explains why increasing the local context window with bigrams or trigrams does not present a larger difference in performance.

Similarly to Georgescul [2010], this work was able to outperform the Wikipedia naïve baseline of 59.01% from table 3.2. This is likely because both systems use some sort of cue selection in order to eliminate spurious cues, which is carried in parameter tuning for the other system. Another observation is that `G3` greatly improves detection for the Wikipedia corpus (+1.65pp compared to `B2`) by restricting considered expressions to annotated cues in training data, providing an extra layer for noise filtering. This suggests that more efficient filters may be able to further improve results.

Finally, it could be observed that F_1 obtained in training may not be used for estimating quality of modality detection. There does not seem to be any correlation between training and testing F_1. It is nevertheless an indicator that optimization alone, without empirically-driven heuristics, may not produce the best results.

3.3.3.4 Error Analysis

There are two main sources of sentence modalization error, one of which is disambiguation. Sets `C` and `D` obtained results poorer than expected for various reasons. Aside from errors from WSD systems, some cues are not covered by usual WSD methods, such as modal verbs. Furthermore, unlike CoNLL-2010 ST systems, disambiguation methods used are not designed specifically for uncertainty. *Sense distribution* is also an issue: if a cue sense produces a number of positive instances that is smaller than θ, then the number of true positives tends to decrease as well.

The other source of error is filtering. Although a very simplistic filtering criteria was used, the algorithm would benefit from a more efficient approach in order to better handle differentiation of infrequent ambiguous cues and noise, as observed in `G3`. Needless to mention, an improvement in filtering should address the sense distribution problem as well, aside from decreasing the likelihood of overfitting.

Finally, it is observed that most errors are caused by WSD, since filtering affects only a small number of instances. In fact, precision is affected exclusively by WSD, as filtering does not cause false positives.

3.3.4 Manual Annotation

Since cue selection is concerned with automatically choosing cue expressions, data annotation consists of only identifying sentence-level modal value. Given the annotation cost in equation (3.2), $\mathcal{C}$ is thus decreased as given by the equation

$$\mathcal{C}(L) = \sum_{L_i}^{L} \sum_{M_j}^{M} T_{modal} \cdot |S| \cdot A(L_i, M_j) \tag{3.10}$$

where T_{modal} is the average time to manually determine if a sentence has a non-zero modality value (complexity of $\mathcal{O}(2)$) and $|S|$ is the total number of sentences in the corpus.

The term $T_{cue} \cdot |W|$ in equation (3.2) is substituted by $T_{modal} \cdot |S|$ in equation (3.10). It is observed that the total time of annotation greatly decreases, because $|S| \ll |W|$ while T_{cue} and T_{modal} have the same complexity. The cost of annotators $A(L_i, M_j)$ also decreases, because it is easier for an annotator to determine modalized sentences, rather than identifying the exact word which causes modalization in the sentence.

It is also added that, since cue expressions do not need to be annotated, sentence-level modality annotation for one language can be reused for other languages by using sentence alignment techniques. The cost would then be further reduced to

$$\mathcal{C} = \sum_{M_j}^{M} T_{modal} \cdot |S| \cdot A(M_j) \tag{3.11}$$

As for the cost versus quality trade-off in the modality domain, the cost is divided by the number of languages and by the average number of expressions per sentence, whereas quality is determined by the performance of selection, which in turn is determined by factors such as quality of disambiguation and cue-to-noise ratio. Nevertheless, promising results have been shown for the proposed method under two different corpora.

3.4 Discussions and Conclusion

Cue selection of modality cue expressions was investigated, proposing a simple optimization algorithm which runs in subquadratic time. It was empirically shown that optimization outperformed state-of-the-art systems in the general case, especially for corpora which are less reliable from a machine learner point-of-view. This suggests an approach with high applicability, suited for other languages or modality types.

One of the advantadges of decoupling cue selection is that cue expression annotation is not required. In addition, high performance was obtained even when using only lemmas and POS tags, suggesting that resource-intensive disambiguation is not required under this approach. The main disadvantadge is that algorithm input only allows tokens such as disambiguated n-grams, and does not support overly complex data structures such as trees. However, it may be argued that the bag-of-words approach is able to satisfactorily model the modalization mechanism.

On annotation cost, it was reduced from word-by-word cue expression to only sentence-by-sentence modal value annotation. The cost of annotators was also decreased, as annotators are not expected to identify cue expressions. Hence, annotation is simpler, more trivial and less prone to inconsistencies. It was also stated that sentence alignment techniques should make propagation of annotation to other languages more trivial.

Finally, some possible improvement points for the method are discussed. The first point is to find a better filter for noise, which is currently done using a minimum frequency threshold θ. The second point is the improvement of cue expression disambiguation by using feature sets and techniques similar to systems in the CoNLL-2010 Shared Task. The investigation of semantically richer feature types is also necessary.

Chapter 4

Tense

4.1 Knowledge Representation

Tense is the grammatical category which locates events in time. It is expressed at two different levels:

- Morphological level: Concerns language-specific grammaticality of verb usage, and is described by conjugation, modal verbs, etc.
- Semantic level: Concerns human perception on the temporality of uttered events, and is described by the logical temporal structure of the event

In order to avoid ambiguity of the term "tense", the first case is denominated herein as *tense morphology*, *morphological-tense* or *m-tense*, and the second as *tense semantics*, *semantic-tense* or *s-tense*.

Tense semantics was initially described by two temporal markers and relations among them, as surveyed by Binnick [1991]. The temporal markers are the event `E` and the time of speech `S`, also referred to as time of utterance. The relations are precedence "`-`" and concurrence "`,`". This allows the definition of three *absolute s-tenses*, namely present (`S,E`), past (`E-S`) and future (`S-E`), as illustrated below:

(4.1) a. He *left* yesterday. `E-S`

b. He *goes* there everyday. `S,E`

c. He *will leave* tomorrow. `S-E`

It was noticed that this simplistic representation lacks descriptive power for cases such as English's present perfect, among others. Other theories, which address this shortcoming, have been proposed and are detailed in the following sections.

4.1.1 Reichenbach's Framework

The arguably most widely accepted theory for s-tense is Reichenbach's framework [Reichenbach, 1947] because of its representational power and simplicity. It defines a third temporal marker, the extra-linguistic reference point `R`. Tense semantics is then described in terms of the relations between `S` and `R`, and between `R` and `E`.

The aforementioned problem regarding the inability to, for example, differentiate English's simple past from present perfect is thus addressed. Markers `R` and `S` share a precedence relation for simple past as illustrated in sentence 4.2a, and a concurrence relation for present perfect as illustrated in sentence 4.2b.

(4.2) a. He *left* yesterday. `R,E-S`

b. He *has* already *left* yesterday. `E-S,R`

The introduction of the reference point allows the definition of six *relative s-tenses*, namely anterior past (`E-R-S`), anterior present (`E-S,R`), anterior future (`S-E-R`), posterior past (`R-E-S`), posterior present (`S,R-E`) and posterior future (`S-R-E`), in addition to the previously mentioned three absolute s-tenses. A complete view of all nine s-tenses is given in table 4.1. This framework also has a number of extensions, such as the usage of a reference interval instead of a point [Dowty, 1979] and multiple reference points [Comrie, 1985].

TABLE 4.1: Representation of s-tense given by Reichenbach's framework

	Past: `R-S`	Present: `S,R`	Future: `S-R`
Anterior: `E-R`	He*'d left* when I arrived `E-R-S`	He *has* already *left* `E-S,R`	He*'ll have left* when I arrive `S-E-R`
Normal: `R,E`	He *left* yesterday `R,E-S`	He *goes* there everyday `S,R,E`	He*'ll leave* tomorrow `S-R,E`
Posterior: `R-E`	I knew he*'d leave* `R-E-S`	I*'ll leave* now `S,R-E`	He*'ll leave* after I arrive `S-R-E`

4.1.2 Sequence of Tense and TDIP

Aside from a clause's internal tense semantics, it is important to also consider how temporality is perceived throughout the flow of text. This was named *sequence of tense*

(SOT) by Reichenbach, and is observed in the following example, in which grammaticality is observed in sentence 4.3a but not in 4.3b. This is because the latter does not present temporal consistency, since `R` is in the past in the first clause and in the present in the second one.

(4.3) a. Harry left when *Sam arrived.* 1st: `R,E-S`; 2nd: `R,E-S`

b. * Harry left when *Sam arrives.* 1st: `R,E-S`; 2nd: `S,R,E`

The SOT mechanism was observed by Reichenbach [1947], who stated the tendency of `R` to remain unchanged across sequential clauses, and was better formalized by Dowty [1979], who introduced the Temporal Discourse Interpretation Principle (TDIP). This principle explains SOT by stating that:

(P1) `R` is at a time consistent with temporal expressions related to the verb; or

(P2) `R` is at a time which immediately follows the reference time of the previous clause, if no temporal expression is available

Case (P1) is thus denominated herein as *adverbial modification*, whereas case (P2) is denominated *SOT modification*.

4.1.3 Hornstein's Framework

Hornstein [1990] proposed a framework for grammaticality validational of s-tense for English based on Reichenbach's framework and SOT.

Simply stating, considering that the semantic value of tense after adverbial or SOT modifications should be consistent with the semantic value before modifications, this framework proposes derivation constraints that, if broken, indicate ungrammaticality.

Basic Tense Structure is then defined as s-tense without modification. One out of the following six s-tenses are defined as possible basic structures: normal past (`R,E-S`), normal present (`S,R,E`), normal future (`S-R,E`), anterior past (`E-R-S`), anterior present (`E-S,R`) and anterior future (`S-E-R`).

By applying modifications, the basic structures are transformed into Derived Tense Structures (DTSs). The resulting derived structure is compared to the basic one, and ungrammaticality is said not to exist if the following two constraints on s-tense derivation are met:

(C1) All markers (S, R and E) which are concurrent in DTS are also concurrent in the basic structure

(C2) The linear order of markers in the basic structure is kept in DTS

It is observed that constraints are respected in the grammatical example 4.3a, but not in the ungrammatical example 4.3b. In the latter case, S and R are concurrent in the second clause but not in the first, breaking the first constraint.

4.1.4 Priorean and Nominal Tense Logic

Prior [1967] proposed a framework based on modal logic. First and foremost, two operators are defined: P, which stands for "*it was the case that*", and F, which stands for "*it will be the case that*". These operators are shifting operators, working in similarly to the mechanism described by the relation between R and E, but better mathematically formalized.

Given some proposition p and standard boolean operators, s-tense is then described by the grammar

$$\phi ::= \mathtt{p} | \phi \wedge \phi | \neg\phi | \mathtt{P}\phi | \mathtt{F}\phi \tag{4.1}$$

This grammar allows more complex representations of tense than Reichenbach's framework, such as multiple shifting (multiple reference points by Comrie [1985]). However, Priorean logic has one severe limitation, which is the absence of a mechanism for referring to specific times, such as for the positioning of R. This was addressed by Nominal Tense Logic (NTL) [Blackburn, 1994], which proposed the concept of nominals.

Nominals are entities that are true at exactly one moment in time. They are represented by variables i, j, k, etc. By using them, it is possible to use modal logic in order to determine the position of events in the time axis according to some defined timestamp. A complete view of NTL is given in table 4.2, which can be compared to table 4.1.

TABLE 4.2: Representation of s-tense given by Nominal Tense Logic

	Past: $\mathtt{P(i}\wedge\phi\mathtt{)}$	Present: ϕ	Future: $\mathtt{F(i}\wedge\phi\mathtt{)}$
Anterior: Pp	He*'d left* when I arrived P(i ∧ Pp)	He *has* already *left* Pp	He*'ll have left* when I arrive F(i ∧ Pp)
Normal: P	He *left* yesterday P(i ∧ p)	He *goes* there everyday P	He*'ll leave* tomorrow F(i ∧ p)
Posterior: Fp	I knew he*'d leave* P(i ∧ Fp)	I*'ll leave* now Fp	He*'ll leave* after I arrive F(i ∧ Fp)

4.1.5 Other Linguistic Phenomena

The tendency of `R` to remain unchanged across sequential clauses has been observed to fail in some cases. Two of these cases, namely backgrounding and shift of perspective, are presented hereon.

Backgrounding [Hopper, 1979] is the linguistic phenomenon which offers supporting information to a text, being normally linked to adjective and noun clauses [Tomlin, 1985]. Background events are not in the chronological sequential skeleton of discourse. In sentence 4.4, although the adjective clause is `S,R,E` whereas its surrounding clauses are `R,E-S`, ungrammaticality is not observed despite SOT being broken by backgrounding.

(4.4) I talked to John, *who is in charge of the event*, and we agreed on the issue.

Shift of perspective [Binnick, 1991] happens when perspective is changed from the speaker to that of a subject, implicating modification in the position of `S`. It is observed in quoted and free indirect speech. In sentence 4.6, the free indirect speech changes the position of both `S` and `R`, thus not generating ungrammaticality.

(4.5) He said *he is not talking to you.*

4.1.6 Manual Annotation

Independent of the framework chosen, data annotation consists of (i) identifying clause sequence, i.e. parent-children relations among clauses for which SOT holds true; and (ii) identification of s-tense for every clause in the set of clauses N.

For (i) identifying clause sequence, a tree structure needs to be extracted, leading to a complexity of $\mathcal{O}(log|N|)$ for every clause.

For (ii) identification of s-tense for every clause in the set of clauses N, in multiclass classification frameworks such as Reichenbach's and Hornstein's, the complexity is $\mathcal{O}(|C|)$, where $|C|$ is the number of possible s-tenses. For NTL, the complexity is expected to be at least the same as for Reichenbach's and Hornstein's frameworks.

As a result, the total annotation cost $\mathcal{C}$ for this task for a set of languages L is based on equation (1.1) and may be roughly estimated by

$$\mathcal{C}(L) = \sum_{L_i}^{L} (T_S + T_C) \cdot |N| \cdot A(L_i) \tag{4.2}$$

where T_S is the averate time to manually determine the clause tree for one clause pair, and T_C is the average time to manually determine the s-tense for one clause. The cost of specialized annotators $A(L_i)$ is expected to be very high.

Considering automatic clause tree identification, equation (4.2) can be reduced to

$$\mathcal{C}(L) = \sum_{L_i}^{L} T_C \cdot |N| \cdot A(L_i) \tag{4.3}$$

4.2 Semantic Analysis

4.2.1 Problem Statement

The pioneer work for the extraction of tense semantics was done by Moulin & Dumas [1994], who attempted to automatically determine Reichenbach's markers to be used for English-French machine translation. However, due to technical limitations, this interlingual approach was dropped in favor of a transfer-based one, which used properties specific to the language pair in order to perform deeper semantic operations.

The inability to perform interlingual analysis was mapped down to the following three difficulties:

(D1) Determination of sequential clauses (*clause anchoring*)

(D2) Determination of how markers are related with temporal expressions

(D3) Determination and interpretation of `R` in respect to `S`

Difficulty (D1) concerns the identification of the sequence of clauses in which the tendency of s-tense continuity is observed because of SOT. Difficulty (D2) concerns how surrounding temporal expressions modify the positioning of temporal markers. Finally, difficulty (D3) concerns how the extra-linguistic point of reference is perceived in the time axis.

4.2.2 Corpora

A subset of the Brown Corpus [Kučera & Francis, 1967] has been annotated with Reichenbach's markers for this work. It consists of eight texts from each of the following genres: Reportage (news), Belles Lettres (essays, biographies) and Adventure (fiction). The corpus was clause-split using automatic clause boundary detection, which is further

detailed in section 4.2.3.1, resulting in over 6,700 clauses. Markers `S`, `R` and `E` were then annotated for this corpus, with an example given in figure 4.1.

> Social Darwinism was able to stave off (...) (`R,E-S`) | However, in recent decades, (`NONE`) | for what doubtless are multiple reasons, (`S,R,E`) | (...) shift has occurred in both facets of national activity. (`E-S,R`) | A concept of responsibility is in process... (`S,R,E`)

FIGURE 4.1: Example of annotation of Reichenbach markers

For this annotation effort, difficulty in discriminating posterior present (`S,R-E`) and normal future (`S-R,E`) was observed. As a general rule, if the event was identified as relating to the present, posterior present was chosen; otherwise, normal future was chosen. Nevertheless, this differentiation was not strictly enforced.

Imperatives, as well as many of the constructions with "*must*" and "*should*" were assigned posterior present. In addition, most cases of infinitives were assigned posterior s-tenses, gerunds were assigned normal s-tenses and participles were assigned anterior s-tenses.

Table 4.3 provides the number of events in the dataset for each s-tense.

TABLE 4.3: Percentage of clauses for each s-tense in subset of Brown Corpus

	Past: `R-S`	Present: `S,R`	Future: `S-R`
Anterior: `E-R`	3.69%	3.30%	0.03%
Normal: `R,E`	48.97%	24.29%	1.86%
Posterior: `R-E`	8.17%	9.45%	0.25%

Additionally, it was observed that clauses were not placed right after their respective parent in the text (non-linearity) in 26.89% of the cases. In other words, branching of the clause tree and clause inversion were identified in more than one quarter of the cases. Furthermore, regarding SOT, the relations between `S` and `R` and between `R` and `E` remained the same in 59.69% of the cases; only the relation between `S` and `R` remained the same in 24.82%; and neither remained the same in 15.50%.

Finally, another notable annotation work was done by Derczynski & Gaizauskas [2011], who assigned Reichenbach's temporal markers to the WikiWars corpus [Mazur & Dale, 2010]. This effort follows the guidelines of TimeML [Pustejovsky et al., 2003a].

4.2.3 Sequential Classification

This work proposes a method based on sequential classification using temporal expressions as features in order to perform interlingual semantic analysis of s-tense, addressing reviously mentioned difficulties. This analysis task has not been attempted to the best of the author's knowledge. The remainder of section 4.2.3 provides details on the proposed method, and section 4.2.4 provides experiments and results.

4.2.3.1 Clause Anchoring

Clause anchoring determines the sequence of clauses in a text, addressing difficulty (D1) described in section 4.2.1. It is divided into two steps: clause boundary detection and anchoring itself.

Clause boundary detection was the target of the CoNLL-2001 Shared Task [Tjong et al., 2001], which identified the position within the string that delimits two clauses. This work uses the best system from the shared task by Carreras & Màrquez [2003], which obtained an F-value of 84.36%.

For *clause anchoring*, it was observed that: sequential and coordinated clauses are linearly linked; subordinated clauses introduce branches to the tree, since they add new information to text that is independent from previous context; and dialogues and quotation also cause branching, as they disrupt SOT by changing the position of `S`. Furthermore, some clauses such as if-clauses and temporal clauses may have their order inverted.

A bottom-up parsing was then used in order to form the clause tree, taking into consideration these concerns. An example of a clause tree is shown in figure 4.2.

First, a pre-processing step identifies the type of coordination and subordination, assigning categories such as coordination (`Coord`), nominal subordination (`Noun`), adjective subordination (`Adj`) and adverbial subordination (`Adv`), and subcategories such as to-infinity (`To_Inf`) based on the first words of the current clause and the last words of the previous clause, as illustrated in the example below:

(4.6) He has frequently refused (`None`) | (...) to obey local laws (`ADV:TO_INF`) | which he considers unjust... (`ADJ:REL_PRON`)

Given the head clause H and the tail clause T (or T_1 and T_2 in the case of branching, for which there are two tail entities), anchors are defined as a grammar whose production rules are given in one of the following three forms:

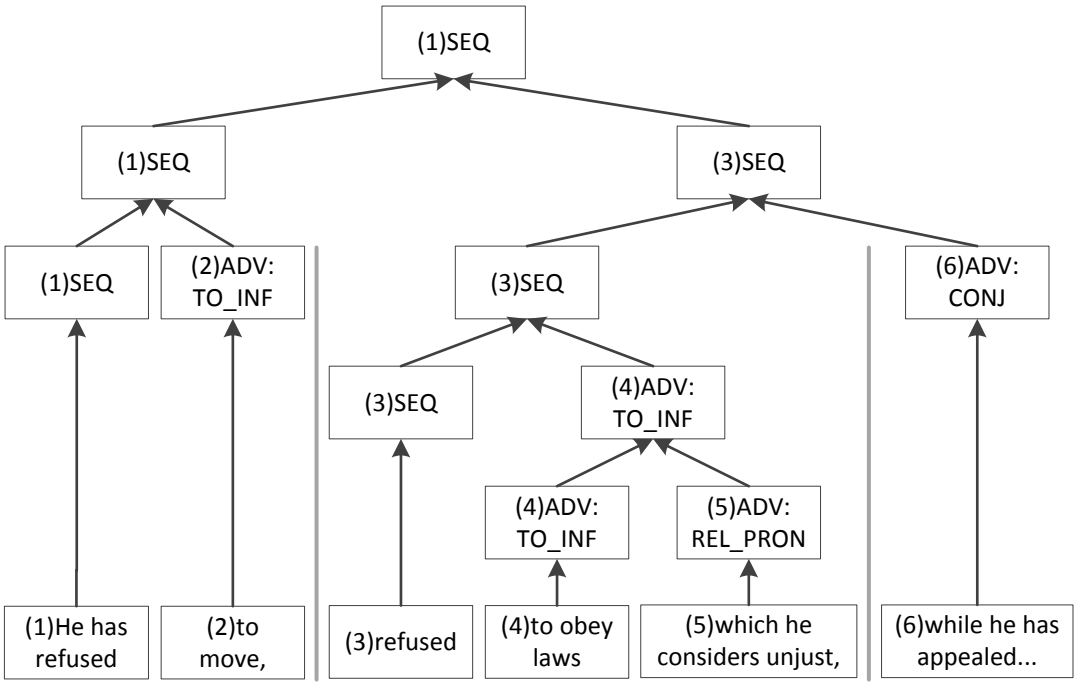

FIGURE 4.2: Example of clause anchoring

- $H \rightarrow HT$ (direct order): "*He has refused* | *to move*"
- $H \rightarrow TH$ (reverse order): "*After the fruit is harvested,* | *it is sold at the market*"
- $H \rightarrow HT_1T_2$ (middle positioning): "*The car,* | *which was red,* | *belonged to him*"

Using these production rules, clauses are then linked in order to form a tree using a parsing algorithm. Shift-reduce has been applied for the extraction of temporal dependencies, such as in ordering events on the time axis [Kolomiyets et al., 2012]. Given $C = (c_1 \dots c_n)$ linear clauses from the text, this algorithm either pushes a clause onto the stack (*shift*) or inversely applies a production rule to the top of the stack (*reduce*). This is done until only one symbol remains in this stack.

In the context of semantic tense analysis, however, the problem of *clause cascading* is observed. Consider a sentence $(c_1c_2c_3c_4)$, which is produced by $c_1 \rightarrow c_1c_2c_4$ and $c_2 \rightarrow c_2c_3$. Before reducing $c_2 \rightarrow c_2c_3$, an algorithm has to analyze if $c_3 \rightarrow c_3c_4$ and $c_3 \rightarrow c_4c_3$ are valid, which is normally solved with n-lookahead. However, due to clause cascading, there is no formal limit to the growth of c_2, which results in no limits to the value n. This translates into difficulty in the decision between shifting or reducing in the algorithm.

A modification of the shift-reduce algorithm is proposed, using a multiple pass approach as shown in algorithm 3.

Algorithm 3 Clause anchoring algorithm

Require: $C = (c_1 \dots c_n)$ linear clauses

```
1: function DEPENDENCY-ANCHORING(C)
2:     while true do
3:         for cursor ← 1 to n do
4:             if !Is-Linked(C, cursor) then Preferential-Link(C, cursor)
5:         Reduce-Farthest-Linked-Clauses(C)
6:         if size(C) = 1 then return C[0]
```

For each pass, the algorithm first moves a cursor to each position (*shift*), linking clauses to the right of the cursor if they are not yet linked. *Preferential linking*, which applies formation rules in a pre-defined order, removes possible rule ambiguities. When the cursor reaches the final position, the algorithm will have linked adjacent clauses to which formation rules could be applied.

Leaf reduction is then performed (*reduce*), removing clauses which have already been linked and which stand farthest away from the root. Consequently, in the previous example, $c_3 \rightarrow c_3c_4$ and $c_2 \rightarrow c_2c_3$ are reduced before $c_1 \rightarrow c_1c_2c_4$, avoiding the problem of cascading for n-lookahead. This process is iterated until only the root is left.

It is observed that punctuation such as commas are used as strong clause delimitors. As a result, in order to simplify the problem, sentences are divided into groups of clauses arranged according to punctuation symbols, such that clauses within the same group are anchored first. This is observed in figure 4.2, which shows three groups of clauses: one for clause (1) and (2); one for clauses (3), (4) and (5); and one for clause (6). The algorithm is applied to each group first, obtaining a group root clause, and is then run again using only the roots of each of these group, finally obtaining the sentence root. The backbone of the text is formed by linking the first clause of the current sentence with the last clause of the sentence that is connected to the root only by NONE and COORD anchors.

4.2.3.2 Feature Selection and Classification

Regarding difficulty (D2) on relating markers and temporal expressions, it is crucial to first and foremost extract such expressions. This task has been extensively researched, being the target of the ACE 2004 TERN Evaluation[1] and the TARSQI project[2] [Verhagen et al., 2005], which addresses timestamping, ordering and reasoning of events,

[1] `http://timex2.mitre.org/tern.html`
[2] `http://www.timeml.org/site/tarsqi/index.html`

automatically annotating text under the TimeML [Pustejovsky et al., 2003a] specification language. In this work, the TARSQI tempex extraction module by Mani & Wilson [2000] is used. It is complemented with a CRF-based extractor, as some temporal expressions such as "*often*" and "*ever*" are not extracted.

In order to model s-tense, other features are extracted using TARSQI and Stanford CoreNLP[3]:

- *Verb m-tense, aspect and modality*: English m-tenses (present, past, future, infinitive, etc.), aspects (perfect, progressive, perfect progressive) and modality (modal verbs) provide information for determining Reichenbach's s-tense categories
- *Verb POS*: Complementary information when verb form is not properly identified
- *Verb lemmas*: Utterance verbs from surrounding clauses are useful for identifying indirect, quoted and free indirect speech clauses
- *Clause links*: Adjective and nominal clauses provide some background information
- *Eventuality types*: A break in the SOT by a eventuality of type "state" is one indication of background independent clause
- *Quotations*: Verbs between quotation marks indicate quoted speech

In many cases, background independent clauses and free indirect speech have no apparent differentiation from regular clauses except for pragmatic information. The extraction of linguistic features for these cases requires further investigation.

The aforementioned features and the extracted temporal expressions are used to address difficulty (D3). They serve as input of a supervised learner, modeling the classification of a clause into one of Reichenbach's categories. The classifier used for this task is CRF [Lafferty et al., 2001], which performs sequential classification, considering the output class of the previous token for determining the current token.

4.2.4 Experiments and Results

4.2.4.1 Experimental Setting

For the corpus presented in section 4.2.2, several settings were compared. The classifier used for the experiments was CRFSuite[4] [Okazaki, 2007], an implementation of CRF.

[3] `http://nlp.stanford.edu/software/corenlp.shtml`
[4] `http://www.chokkan.org/software/crfsuite/`

In addition, a single classifier for all nine s-tenses (simple present, anterior past, etc.) was employed, because it is possible to model the interaction between all three markers, unlike when using separate classifiers for absolute s-tenses (present/past/future, i.e. the relation between `S` and `R`) and relative s-tenses (simple/anterior/posterior, i.e. the relation between `R` and `E`).

Firstly, the effects of sequentiality in classification was evaluated. The input for the CRF classifier was modeled as sequential and non-sequential using automatic clause anchoring, with the non-sequential input expected to be the lower baseline.

Different clause anchoring methods were also compared, namely: the proposed automatic anchoring indicated in section 4.2.3.1; a linear anchoring, considering clauses in the order they appear in the text without any branching; and manual anchoring, which was determined by a human annotator and is expected to be the upper baseline.

A Statistical Machine Translation (SMT) [Koehn, 2010] baseline was also proposed in order to provide a comparison with a shallow linguistic technique. Google Translate[5] was used for translating the corpus from English to Portuguese (EN→PT) and from English to Japanese (EN→JP). The resulting translation was then evaluated in terms of correctness according to Reichenbach's categories. In other words, errors concerning verb choice, passive voice usage, etc. were disregarded. In the example, the first translation 4.7b would be considered correct from the s-tense perspective, whereas the second 4.8b would not.

(4.7) a. EN: He runs everyday. Present

b. JP: Kare wa mainichi jikkō saremasu. Present
(lit. "*He is put into execution everyday*")

(4.8) a. EN: He is running tomorrow. Future

b. * JP: Kare wa ashita jikkō sareteiru. Present continuous
(lit. "*He is being put into execution tomorrow*")

In the open domain, the language pair EN→PT produces satisfactory translation, whereas EN→JP does not. However, in tense translation, accuracy values are 84.0% and 81.8% respectively. Although at first counterintuitive, these values indicate that tense translation is largely dependent on the source language. It is expected that source languages with ambiguous m-tenses according to the target language (ex.: the tenseless Chinese as source and a tense-rich Romance language as target) have lower translation accuracies.

[5] `http://translate.google.com/`

For all classification settings, accuracy and macro-average precision, recall and F-value were used as evaluation metrics. For the SMT baseline, since translated sentences were evaluated based on correctness (correct or incorrect), only accuracy was calculated.

4.2.4.2 Experimental Results

Table 4.4 states the accuracy for each of the previously mentioned settings, comparing unified and separate classifiers. It is observed that using separate classifiers for absolute and relative s-tenses results in lower accuracies for all settings, indicating that they are not able to properly model the relations among markers.

TABLE 4.4: Results of s-tense analysis using Reichenbach's markers

Setting		Result			
SOT	Anchoring	*Acc*(%)	*P*(%)	*R*(%)	*F*(%)
Non-sequential	—	88.5	64.8	57.5	60.9
Sequential	Linear	89.5	65.8	57.7	61.5
Sequential	Automatic	90.8	67.4	61.9	64.5
Sequential	Manual	91.1	67.1	62.2	64.6
SMT Baseline (EN→PT)		84.0	—	—	—
SMT Baseline (EN→JP)		81.8	—	—	—

Moreover, sequential classification produces better results, as it is observed that non-sequential classification produces the lowest F-value of 60.9%. When comparing the three sequential approaches, automatic anchoring obtained results better than those of linear anchoring, and almost as good as manual anchoring (upper baseline). This suggests that the proposed automatic is able to achieve classification whose performance is close to the maximum using a fixed feature set and training data.

Finally, while the differences in accuracy and F-value is not numerically large, it should be noted that there is branching (non-linear clauses) in only 26.89% of the cases, and that there is break in SOT in only 40.31% of the cases, as stated in section 4.2.2.

4.2.4.3 Error Analysis

Most of the obtained errors concern changes in `R` and `S` in cases where there is no explicit context from which to infer the new position of the markers. In example 4.9a below, there is no indication of the simple present in "*No telling*". Other errors occurred because of component failure. In example 4.9b, the shift of perspective was not properly addressed because verb form extraction for the second clause presented an error.

(4.9) a. Mike lifted him (...). | "No telling | how good this horse is."
Obtained: Simple Past (`R,E-S`)
Expected: Simple Present (`S,R,E`)

b. He believed | there are a number of qualified city residents...
Obtained: Posterior Past (`R-E-S`)
Expected: Simple Present (`S,R,E`)

It should be noted that component errors in clause boundary detector are not propagated when verbs and temporal expressions are consistently grouped within the same clause, as s-tense is inherited from the previous clause in 59.69% of the cases due to SOT.

4.3 Alternative Semantic Analysis

4.3.1 Problem Statement

Problems concerning the semantic analysis method presented in the previous section will be discussed hereon. They are roughly divided into two groups, as follows.

The first group concerns several problems related to current s-tense descriptors. It is argued herein that `R` is not intuitive, which increases the cost of annotators and causes inter-annotator and cross-language inconsistencies. Three cases are illustrated hereon:

- Extra-linguistic property of `R`: If native speakers of a given language L may not be able to identify a nuance in `R` if it is not observed in L. One common example is English's simple past (`E,R-S`) and present perfect (`E-R,S`), whose differentiation may not be trivial for non-native speakers. Another example is posterior present and normal future, as observed in the following examples. Cases 4.10a and 4.10b are straightforward, but cases 4.10c and 4.10d are not. In fact, many scholars such as Declerck [1986] have criticized the excessive number of s-tenses, arguing that some of them are not clearly perceived in any natural language.

(4.10) a. I will go now. `S,R-E`
b. I will go tomorrow. `S-R,E`
c. I will go soon.
d. "I will do it", offered the student.

- `R`'s positioning not consistent with observed temporal semantics: Comrie [1985] has criticized the fact that simple past and present perfect are differentiated by

positioning R in the past or present, while in the sentences such as "*I have never gone*", the adverb "*never*" relates to the interval between negative infinity and now, which may be addressed by using a reference interval Dowty [1979]. However, there needs to be a mechanism to avoid ambiguity as in sentences 4.11a and 4.11b. In both cases, the event happens within the reference interval, described by the left-most point R_0 (negative infinity in both cases) and by the right-most point R_1 ("*you came*" in 4.11a and "*7pm*" in 4.11b).

(4.11) a. He had left before you came. R_0-E-R_1-S

b. He left before 7pm. R_0-E-R_1-S

- R's behavior not consistent in different situations: On one hand, when in the present, R indicates some temporal relation to now. On the other hand, when in the past or future, R serves as a pivot away from which E is placed, ex.: past in the future for 4.12a and future in the future for 4.12b. Furthermore, R is not a pivot in 4.12c, since "*after tomorrow*" simply provides an imprecise temporal location for the future event.

(4.12) a. I will have finished by tomorrow. S-E-R

b. I will go after you arrive. S-R-E

c. I will go after tomorrow. S-R,E

In order to address this, a novel description system is proposed. It aims to explain and better formalize s-tense with non-extra-linguistic descriptors which are known for speakers of different languages and whose representational power is at least the same as Reichenbach's markers, hence making manual annotation process more intuitive even for non-specialists.

The second group concerns the total cost of annotation for a corpus. Because of the high level of annotator specialization required, the cost of annotators per unit of time A in equation 4.3 is expected to be high as well. It is observed that, due to processing of tense semantics still being an unexplored area, works on s-tense annotation are limited to the full corpus annotation effort by Derczynski & Gaizauskas [2011] and this research, to the best of the author's knowledge.

In addition, cost is also influenced by language specificity of syntactic constructions, which may cause cross-language inconsistencies. For example, in sentence 4.13, the verb complex can be considered as one entity, in which the future action "*run*" is the base proposition modified by a volitive modality, or as two entities, in which the state "*want*" is considered separately from "*run*".

(4.13) I want to run tomorrow.

This work addresses this problem by exploring automatic inference of s-tense, compositing it from other semantic elements of the clause. The idea is to provide theoretical grounding for inference from verb conjugation, temporal expressions, SOT and other linguistic features.

4.3.2 Composite Temporal Structure and Semantic Composition

This work proposes an NLP-motivated theoretical linguistic framework for addressing problems concerning annotation of s-tense. This approach was motivated by the fact that current theory presents shortcomings from a manual annotation perspective, as well as for not providing proper support to s-tense inference. This is because semantic theories are concerned with explaining the linguistic phenomena, assuming semantics to be known, whereas s-tense analysis is concerned with extracing semantics assuming phenomena to be correct.

Unlike other theories, the proposed description system describes all temporal entities (m-tenses, adverbial and SOT modifiers) outside of context, which in turn enables annotation of temporality on a per-expression basis (centralized dictionary annotation) rather than a per-occurrence basis (decentralized full corpus annotation). Each sense from the dictionary annotation has a semantic structure described in terms of this description system, denominated *Basic Temporal Structure* (BTS). Consequently, each uncontextualized temporal entity is explained by a set of BTSs. By centralizing definitions, the management of the annotation process and correction of inconsistencies are facilitated.

The BTSs of m-tenses, adverbial modifiers and SOT modifiers relevant to a clause are then combined into this clause's *Composite Temporal Structure* (CTS). A CTS is the semantic structure for a clause's s-tense in this theory. The process of combination, denominated *semantic composition*, is responsible for choosing the correct BTS according to the context. In other words, semantic analysis is modeled as defining temporality of various entities in terms of s-tense descriptors, and combining them.

CTSs build on the idea introduced by Hornstein and NTL. Hornstein considers s-tense as a derivation of its default uncontextualized value (basic structure), which is nevertheless rudimentary for practical purposes; NTL considers s-tense as a combination of contextualized temporal entities, but there is no mechanism for disambiguation of propositions, and no internal temporal structure is defined for nominals. CTSs, on the other

hand, are the combination of uncontextualized entities, unlike Hornstein (derivation of uncontextualized entities) and NTL (combination of contextualized entities).

Compositionality of s-tense has been discussed by Musan [2001], concluding that s-tense is not compositional on the syntax-semantics interface, ex.: "*have*" does not carry semantics for English's perfect. However, when considering s-tense a property of the clause and concerning compositionality in terms of semantic elements, it is argued herein that s-tense is compositional, ex.: one of the senses for "*never*" carries semantics for s-tense anteriority. This compositional behavior has also been observed in NTL, which mathematically formulates how propositions and nominals combine into a clause's s-tense.

When concerning annotation costs, the adoption of other approaches are discussed. Sampling-based methods including active learning are difficult to apply, since SOT reference might be lost in the process. One possibility would be to use clauses with adverbial modification, since these are points in the clause tree in which SOT modification may not take place; however, unlike stated by the TDIP, adverbial and SOT modifications are not mutually exclusive, as observed in "*He said he might come on Friday*" (past reference in inherited from the first clause despite the existence of an adverb), aside from the fact that the number of adverbial modifications is small. In addition, semi-automatic annotation methods are currently not possible, due to the lack of a supporting theory for s-tense inference. Finally, computer-assisted annotation would be possible by identifying the temporal adverbials and the parent clause's s-tense, but its impact on the final cost is low.

4.3.2.1 Inspection, Focusing and Shifting

The non-extra-linguistic operators used for describing BTSs and CTSs are introduced in this section. They are named inspection, focusing and shifting.

Inspection refers to time intervals that are analyzed in order to assess the truth value of an event or any other temporal entity at a given moment. For example, the inspected interval for "*I have [never] gone*" equals to all moments from negative infinity to now, which can be described in terms of the first inspection point t_0 (equals to $-\infty$ in this case) and last inspection point t_1 (equals to 0). Inspection is an approach based on the idea by Guéron [2007] that the speaker moves a cursor along the time axis in order to define an event's location. It differs from the reference interval by Dowty [1979] in the sense that it provides a concrete action for determining the interval to be considered.

Focusing refers to points within the inspected intervals that are linguistic identifiable. For example, the event "*come*" is a focus point in "*before you came*", whereas "*now*"

is an implicit focus point in the sentence "*I am running*". Focus points, together with shifting, solve possible ambiguities caused by using intervals as in 4.11a and 4.11b. Although not trivial to obtain, focusing is a formality for theoretical soundness that does not need to be analyzed in practical situations.

Shifting refers to forcing the placement of a temporal entity in a position on the time axis away from a focus point, similarly to Prior's `P` and `F`. For example, "*before*" shifts an event towards the past in "*before you came*".

These descriptors present a novel representation system which describes temporal entities using tense descriptors. Because this system is more semantically granular, they may be mapped to Reichenbach's markers, although the opposite is not necessarily true. Table 4.5 summarizes how this system compares to Reichenbach's markers and NTL when considering a single `R` punctual in time. The points t_0 and t_1 are used for describing the relations between `S` and `R` and between `S,R` and `E`, while shifts are used for the relations between `R-S` and `E` and between `S-R` and `E`. It is thus noted that anterior/posterior present is explained by inspection alone, while anterior/posterior past/future is explained by shifting, solving one of the semantic inconsistencies found in Reichenbach's framework.

TABLE 4.5: Representation of s-tense given by inspection and shifting

	Past	Present	Future
Anterior	He*'d left* when I arrived $t_0 < 0$, $t_1 < 0$, past shift	He *has* already *left* $t_0 < 0$, $t_1 = 0$, no shift	He*'ll have left* when I arrive $t_0 > 0$, $t_1 > 0$, past shift
Normal	He *left* yesterday $t_0 < 0$, $t_1 < 0$, no shift	He *goes* there everyday $t_0 < 0$, $t_1 > 0$, no shift	He*'ll leave* tomorrow $t_0 > 0$, $t_1 > 0$, no shift
Posterior	I knew he*'d leave* $t_0 < 0$, $t_1 < 0$, fut. shift	I*'ll leave* now $t_0 = 0$, $t_1 > 0$, no shift	He*'ll leave* after I arrive $t_0 > 0$, $t_1 > 0$, fut. shift

4.3.2.2 Basic Temporal Structures

Definition

BTS is the temporal structure of one sense of a temporal element. It provides s-tense representation by means of a triplet (t_0, t_1, s). Possible values for the first inspection point t_0 are $t_0 < 0$, $t_0 = 0$, $t_0 > 0$ or $t_0 =?$; for the last inspection point t_1 are $t_1 < 0$, $t_1 = 0$, $t_1 > 0$ or $t_1 =?$; and for the shifting operator s are $s = p$ (past shift), $s = n$ (no shift) or $s = f$ (future shift). The values $t_0 =?$ and $t_1 =?$ indicate incomplete s-tense information. The examples below illustrate how multiple BTSs (temporal semantic structures) of raw temporal entities are annotated:

(4.14) would go

a. habit in past: $(t_0 < 0, t_1 < 0, s = n)$

b. future in past: $(t_0 < 0, t_1 < 0, s = f)$

c. polite request: $(t_0 = 0, t_1 > 0, s = n)$

(4.15) on Tuesday

a. some past Tuesday: $(t_0 < 0, t_1 < 0, s = n)$

b. some future Tuesday: $(t_0 > 0, t_1 > 0, s = n)$

(4.16) before

a. conjunction: $(t_0 =?, t_1 =?, s = p)$

b. preposition: $(t_0 =?, t_1 =?, s = n \text{ or } s = p)$

Each uncontextualized raw form of a temporal entity may be described by a set of one or more BTS triplets. It is possible to generalize these raw forms, ex.: case 4.14 may be generalized as all non-stative verbs in the form "would + [infinitive]". As a result, while Reichenbach's and Hornstein's works require markers to be annotated for every single clause because of their context-sensitiveness, this research enables annotation on generalized raw forms; most of the burden of the analysis process falls onto semantic composition, which is automated.

BTS triplets compose the proposed structure of tense semantics. Annotating these descriptors in a text is easier than annotating `R`, because the task is simplified to locating two linguistic-identifiable points in time (for inspection points) and establishing a before/after relation (for shifts). In addition, since only a small number of generalized forms need to be annotated, it is easier to guarantee consistency because on centralization of annotation. Finally, the problem of determining the relation between `S` and `R` becomes the problem of identifying the position of t_0 and t_1 relative to now. This may still be difficult, since `S` is punctual and is constantly changing its position on the time axis. An idea is to consider an infinitesimal interval around now, ex.: if it can be expected that the event "*He will do it*" is performed in the subsequent moments, then $t_0 = 0$; otherwise, $t_0 > 0$.

A study of the behavior of inspection, focusing and shifting operators for BTSs is given hereon.

BTSs of verbs

For inspection, verbs always define at least one inspected interval, since they are responsible for expressing events in time. However, the boundaries of such intervals are not precisely stated, as verbs do not possess a mechanism for absolute positioning, unlike adverbial modifiers.

For shifting, m-tenses are classified as *not shifted* (ex.: English's simple present); *always shifted* (ex.: English's past perfect); and *occasionally shifted*, which is shifted only upon proper modification, not being shifted by default (ex.: English's future, as in sentence 4.12b).

For focusing, always-shifted m-tenses introduce an implicit focus point, which may be reinforced by adverbial phrases, ex.: "*He had talked*" happens before some event not stated in the text. Focus points are not observed for not- or occasionally-shifted m-tenses. In the latter, adverbial modification is responsible for focusing.

Finally, aspect and modality also influence the mapping of m-tenses into s-tenses. For example, present is introduced by an adverb in 4.17a and by the gnomic aspect in 4.17b. The irrealis mood adds a meaning of unrealized future in the first clause of 4.17c.

(4.17)
a. I go to school everyday. $(t_0 < 0, t_1 > 0, s = n)$
b. The sun rises in the East. $(t_0 < 0, t_1 > 0, s = n)$
c. If I go there [, he will get mad at me.] $(t_0 = 0, t_1 > 0, s = n)$

BTSs of adverbial modifiers

Temporal adverbial modifiers can be classified based on anchoring [Smith, 1978]. An anchored modifier has an explicit relation with the speech point, referring to a moment either before, after or simultaneous to `S`. As a result, the inspected interval is clearly defined. For example, "*yesterday*" (past), "*ago*" (past) and "*tomorrow*" (future) are anchored adverbial phrases, while "*on Tuesday*", "*in April*" and "*this Monday*" are not, since they may refer to both past and future without context, as seen in example 4.16.

Adding to this initial notion of anchoring, it is stated herein that even anchored modifiers may be ambiguous when the inspected interval is smaller than the interval indicated by the adverb. This will be referred to as *sub-interval*. For example, "*today*" indicates a 24-hour period, but the inspected interval for "*I worked today*" ends before now, and the interval for "*I will work today*" starts after now.

Another consideration is when there is more than one adverbial modifier. Although "*tomorrow*" is not shifted, it is linked to an event shifted by the adverbial clause "*after*

you return" in "*I will leave tomorrow after you return*". This indicates that all adverbial modifiers must be evaluated beforehand.

Finally, adverbial modifiers may be explicitly stated only in other clauses, as in the parallelization "*He comes home, sits on the sofa and watches TV everyday*". In such cases, the scope of adverbials should be properly analyzed in surrounding clauses.

BTSs of SOT modifiers

Assuming that the CTS from the parent clause is known, it is used in order to create the BTS for SOT modification. There are therefore two concerns. The first is the identification of the parent clause from which s-tense is inherited. This is solved using clause anchoring efforts, such as the one presented in section 4.2.3.1.

The other is the inheritance mechanism. It determines what information of the parent's CTS is to be used as the SOT modifier's BTS. Possible cases for inheritance are exemplified below. In 4.18a, inspection and shifting are inherited. In 4.18b, only inspection is inherited. In 4.18c, nothing is inherited. In 4.18d, inspection is partially inherited: t_1 is maintained in the future, whereas t_0 goes from past to now. In terms of Reichenbach's framework, this concerns inheritance from normal to anterior/posterior present, and is similar to case 4.18b.

(4.18)
a. He will have finished by then; she won't [have finished].
b. He will have finished by then. This will make her happy.
c. He tried, but he will have to do it tomorrow.
d. He is always trying, but will not succeed.

4.3.2.3 Composite Temporal Structures

Similarly to BTSs, CTSs are described by the triplet (t_0, t_1, s), except that $t_0 =?$ and $t_1 =?$ are not accepted. Semantic composition is then defined as the process which takes several BTSs related to an event (one BTS for each sense of the verb, zero or more BTSs for adverb modifiers, and zero or one BTSs for the SOT modifier), disambiguates them, and elects the descriptors which form the CTS. Figure 4.3 illustrates BTSs for the present continuous of "*run*" and for the adverb "*always*", showing two pairs of BTSs that are consistent, namely $(t_0 < 0, t_1 > 0, s = n)$ and $(t_0 > 0, t_1 > 0, s = n)$. It is observed that ambiguity still persists in this case, because there is no mechanism to account for context.

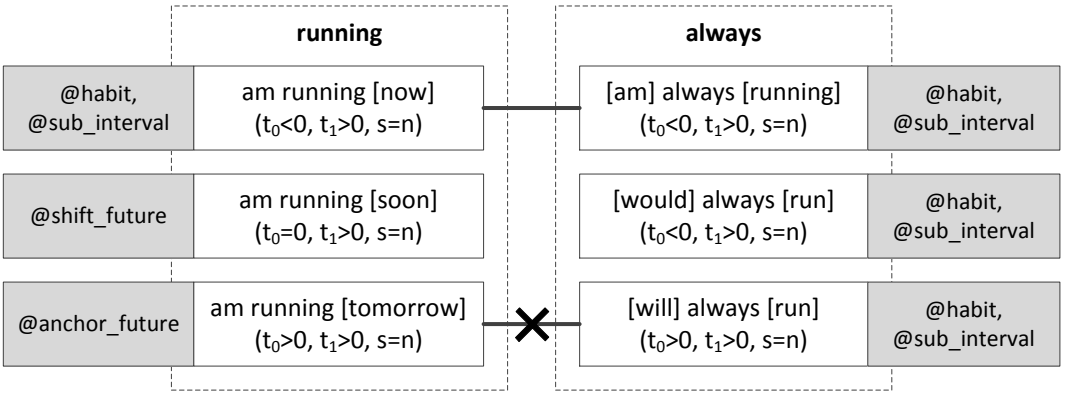

FIGURE 4.3: Example of semantic composition using connectors

Connectors are then used for accounting for context in order to solve ambiguity. They are defined as attributes of BTSs, such that composition between two BTSs is only allowed when both possess at least one common connector. In the example, since the top pair of BTSs possesses common connectors `@habit` (for habitual events) and `@sub_interval` (for adverbial modifiers whose inspected interval is smaller than the denoted interval), they are composed, unlike the bottom pair. The usage of connectors is a simple approach to be used for automatic composition, solving ambiguity by carrying some information on word form. It should also be noted that different connectors may be used for different languages, since connectors are not part of the CTS and are not relevant for cross-language applications.

Figure 4.4 summarizes the previously raised points raised. When there is ambiguity, composition should choose proper descriptors for CTS by matching BTSs and their connectors. The usage of a default BTS is possible if ambiguity is not solved after all composition steps; however, this should be employed with caution in order not to generate errors.

4.3.3 Experiments and Results

4.3.3.1 Experimental Setting

A proof of concept of automatic inference of s-tense using the proposed theory is evaluated hereon in terms of capabilities and limitations. By automatically inferring CTS, it is possible to annotate only verbs' BTSs, adverbial modifiers' BTSs and ambiguous CTSs, instead of annotating every clause. The ultimate objective of automatic inference is to completely avoid errors, which cannot be differentiated from correct cases unless data has been previously annotated, while decreasing ambiguity on a best-effort basis, since ambiguous cases can be easily identified, but need to be manually annotated.

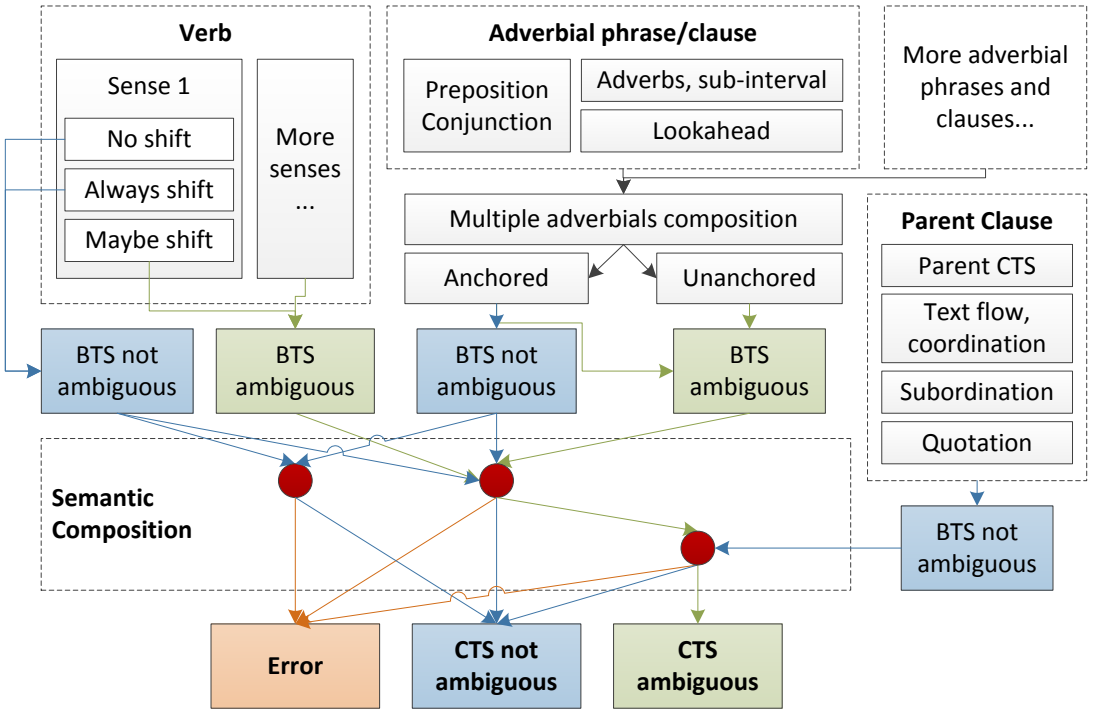

FIGURE 4.4: Overview of semantic composition

The corpus used in this research is based on the Brown Corpus subset as described in section 4.2.2. For this experiments, the expected CTS for all events were annotated, and unlike the original corpus, cases such as "*I want to run tomorrow*" (example 4.13) were considered as separate entities. This resulted in a total of 13,259 events.

Linguistic information was extracted in order to perform inference as follows. POS tags were automatically extracted and manually corrected because of errors such as misclassifying VBD (simple past) and VBN (past participle); clause trees were automatically determined as indicated in section 4.2.3.1 and manually corrected; temporal expressions were automatically extracted as indicated in section 4.2.3.2; stative aspect was automatically classified [Saurí et al., 2005] and manually corrected; and realis/irrealis modality was manually assigned.

This work assigned BTSs and connectors for each sense of all temporal entities within the corpus. The annotation process is described hereon.

First, each clause was represented by a list of tags, which indicate linguistic features found within the clause and are used for facilitating the annotation process. For verbs, tags were derived from lemma and POS tags within the verbal complex (ex.: `#past` from VBD, `#perfect` from "have" + VBN); tags `#infinitive` and `#imperative` were manually assigned from VB; and tags `#irrealis` and `#state` were also used. For adverb modification, lemmas of adverbial temporal expressions were used for generating tags

such as `#already`, `#currently` and `#ago`; prepositions, conjunctions and noun expressions were manually annotated, as this was faster than devising an automatic extractor due to the small number of instances. For SOT modification, tags for parents' CTSs were automatically added during inference. CTSs were encoded as tags representing one of Reichenbach's nine s-tenses. Tags for coordination (`#coord`) and first clause of a quotation (`#quote`) were also considered.

After tags were identified, one BTS was assigned for each sense of each temporal entity represented by a tag or group of tags. For example, three different BTSs, namely $(t_0 = 0, t_1 > 0, s = n)$, $(t_0 > 0, t_1 > 0, s = n)$ and $(t_0 > 0, t_1 > 0, s = f)$, were assigned to the tag `#will`. Connectors were then defined for adverbial and SOT modifiers, with one or more connectors being assigned to each BTS. Adverbial modification connectors include `@habit` (assigned to tags such as `#frequently`), `@sub_interval` (`#always`), `@anchor_past` (`#ago`), `@shift_past` (`#before`), etc.; SOT connectors include `@parent`, when both inspection and shifting are inherited from the parent clause, and `@sot`, when only inspection is inherited.

4.3.3.2 Result and Error Analysis

From the 13,259 events, the best-effort automatic inference was able to reduce ambiguity to 169 events and errors to 8 events. A summary on the causes of ambiguity and error is then given.

Ambiguity was forced during BTS assignment of cases which are difficult to guarantee a correct judgement using available information. This in turn forces manual annotation. Cases for tags `#past` (80 cases), `#will` (2), `#would` (13), `#could` (7) and `#gerund` (27) fall into this category. For example, in sentence 4.19a, ambiguity is forced because it is difficult to decide whether there is a past shift, as English's simple past is commonly used instead of both present perfect and past perfect m-tenses. In sentence 4.19b, '*would*" belongs to the first verbal clause of the subtree branch created by the quotation marks. This causes ambiguity because s-tense may either start a new temporal axis or be inherited either from its parent clause.

(4.19) a. [...] in the "coolest election I ever saw in this county".

b. "Never mind", she said sternly. "It would not matter [...]".

When not forced, ambiguity is caused when more than one verb BTSs are composed (resulting in multiple CTSs) or when none of multiple BTSs is composed (it is impossible to judge which BTS is correct). These are the cases for `#present` (38 cases) and `#might`

(1). In sentence 4.20, the two ambiguous senses ($t_0 < 0, t_1 > 0, s = n$) and ($t_0 = 0, t_1 > 0, s = n$) of English's present continuous were not composed because of the absence of adverbials.

(4.20) Douglas [...] said the transit company is reviewing the work on...

The first main source of error is when SOT information is not within the parent clause; it is implicit or is in other clauses, as for tag `#would` (4). In sentence 4.21a, the narration previously mentioned "*go off*" as a past event, but this information is not explicit in this clause. Likewise, the past reference in sentence 4.21b comes from the noun "*announcement*" and not from the parent clause.

(4.21) a. [...] I do not savvy why you'd go off.

b. [...] The announcement that the city would sue for recovery...

Error was also caused by the verb "*come up*", which has an implicit future reference in 4.22 for the tag `#gerund` (1). While ($t_0 < 0, t_1 > 0, s = n$) was obtained because of the gerund, ($t_0 = 0, t_1 > 0, s = n$) was expected.

(4.22) [...] with Arkansas coming up Saturday.

The BTSs used in this proof of concept were not able to completely handle ungrammaticality, as seen in 4.23 for tag `#can` (1). The incorrect "*can*" should be replaced with the correct "*will be able to*" in order to avoid inference error. However, expected ungrammaticality was properly handled, as previously stated for past simple, present perfect and past perfect in sentences 4.19a and 4.19b.

(4.23) If any of us miss, they can pick up the pieces. (sic)

Finally, ambiguity for tag `#participle` (1) and error for tags `#infinitive` (1) and `#could` (1) were observed in the first clause of a given text.

4.3.4 Manual Annotation

For automatic inference, manual annotation consists of assigning BTSs and connectors to verb conjugations, temporal expressions and SOT modifiers in an uncontextualized

manner, and to annotate the CTSs of ambiguous cases, with the rest of the process being performed automatically.

Given the annotation cost in equations (4.2) and 4.3, $\mathcal{C}$ is decreased as follows

$$\mathcal{C}(L) = \sum_{L_i}^{L} (T_{BTS} + T_C \cdot |N_{amb}|) \cdot A(L_i) \tag{4.4}$$

where T_{BTS} is the time to manually determine all BTSs and connectors ($T_{BTS} \ll T_C \cdot |N|$), and $|N_{amb}|$ is the total number of ambiguous clauses in the corpus ($|N_{amb}| \ll |N|$).

It also observed with this approach that the very high cost of annotators $A(L_i)$ decreases considerably, as the novel operators do not require abstract knowledge on extra-linguistic components.

As for the cost versus quality trade-off, cost is drastically decreased, whereas quality is determined by the number of errors in the outcome; correctly annotated instances using automatic inference is the same as when performing full corpus annotation. While errors might be difficult to map without analyzing the entire corpus, it is observed that it is difficult to guarantee a small number of inconsistencies when using Reichenbach's markers as the annotation schema.

4.4 Discussions and Conclusion

First and foremost, this work investigated the semantic analysis task for s-tense, addressing previously identified difficulties by proposing a clause anchoring algorithm, and using temporal expressions as linguistic features as well as the sequential classifier CRF in order to model SOT. The proposed approach obtained satisfactory results, even though some linguistic phenomena such as backgrounding and shift of perspective were not completely modeled by currently available features.

The alternative analysis was also investigated. A theoretical approach was taken because state-of-the-art linguistic theories lacked the mechanism to explain the abstraction of s-tense from verbs, adverbs and context. The proposed theory: (1) introduced the operators of inspection, focusing and shifting, which are more intuitive for annotators; (2) studied how these operators were perceived for temporal entities; and (3) studied the mechanism through which they combine. This composition mechanism was also evaluated in a proof of concept of automatic inference of tense semantics.

On annotation cost, the complexity was reduced from full corpus to BTS annotation and ambiguous cases only. In addition, the usage of better formalized, more intuitive descriptors is expected to decrease the cost of annotators.

Finally, several points observed from the proof of concept are discussed. The first point is on the lack of semantic information for some cases, indicated by 4.21a, 4.21b and 4.22. Case 4.21a requires contextual information not retrieved by SOT; 4.21b requires contextual information encoded in words other than verbs and adverbial modifiers, which was also observed by Nordlinger & Sadler [2004]; and 4.22 requires internal semantic information of specific constructions. The first two cases can be modeled as a new type of modification, whereas the latter would need to be addressed by raw form annotation using a dictionary.

The second point is on the first event of a trunk – i.e. the first clause of a entire text or the first clause of a quote. For the first clause of a text, it is not uncommon for authors, in an attempt to make texts more interesting, to produce first events that might be difficult for automatic inference. As for the first clause of a quote, there is no apparent rule indicating when the position of `S` is actually modified. The simplest way to address this is by manually annotating such cases, since they are easily identifiable.

The third point is on the detection of ungrammatical sentences. The proposed implementation of autmatic inference was designed as a proof of concept whose main objective is to decrease annotation costs. As a result, only some commonly occurring cases of ungrammaticality (ex.: using simple past instead of present perfect, such as in 4.19a) were coded. Using the proposed semantic composition for error detection is yet to be investigated.

The fourth point is on the linguistic features used, as presented in section 4.3.3.1. This proof of concept used manually annotated or manually corrected features, which is also the case for most annotation efforts. However, automatic inference under less reliable environments, as well as methods for increasing its performance, is an interesting area of research for analytical Semantic Computing.

Finally, it was noticed that there is no NLP tool which reliably classifies aspect, which is an interesting research area for interlingual analysis, and realis/irrealis modality, which is analyzable by means of the pragmatic element of assertion.

Chapter 5

Discussions and Conclusion

5.1 Annotation Cost Effectiveness

This research showed linguistically-motivated computational methods for providing more cost effective approaches to manual annotation for different domains in Semantics. Cost reduction and consequent quality trade-off for each case are discussed in this section, showing how cost of the annotation process is affected by human-computer interaction.

First and foremost, some factors that affect estimation of annotation cost are discussed. Number of entities, number of corrections, order of examples, nativeness of annotator, among others [Settles et al., 2008; Ringger et al., 2008], as well as combination of instance selection, annotator and annotation task [Arora et al., 2009] have been investigated for the tasks of named entity recognition, image retrieval and information extraction. Cost estimators based on fixed sets of factors were proposed, with known drawbacks being, for example, the inability to determine an estimate for new annotators or new user interfaces.

In addition, more difficult annotation tasks such as deep semantic ones possess a steeper learning curve. The time to annotate one instance is different depending on how accustomed the annotator is to the task, and the steepness of the curve differs among individual annotators. Hence, a proper estimator should determine parameters for the curve pertaining to each annotator. This is outside of the scope of this research, but is nevertheless an interesting topic of research.

This work compares annotation methods qualitatively hereon due to the observed difficulty to properly estimate cost. Quantified comparisons are given whenever possible.

For the semantic domain of contextual semantic relations, full corpus annotation and bootstrapped set expansion are compared. Co-training, which reportedly does not

present significant improvement, and active learning, which is not suitable under settings with class imbalance, are not considered at this stage. When comparing full corpus annotation with set expansion, it is firstly observed that the former requires a multiclass decision whereas the latter requires a binary decision. In the full corpus example 5.1, the annotator would have to analyze which of the 46 CDL relation classes (see appendix D) apply to each case.

(5.1) Determining the class for the relation in each sentence:

a. "*Mary*"→"*walk*" in "*John walked with Mary*" Co-Agent (`cag`)

b. "*Mary*"→"*share*" in "*John shared with Mary*" Partner (`ptn`)

On the other hand, in example 5.2, given that set expansion is currently being held for the Co-Agent (`cag`) class, only a true or false label must be assigned. The decision cost is thus considerably lower than the above approach (23 times less at maximum), although the familiarity of the annotator with the task might decrease this difference.

(5.2) Determining if relation is of class Co-Agent (`cag`) in each sentence:

a. "*Mary*"→"*walk*" in "*John walked with Mary*" True

b. "*Mary*"→"*share*" in "*John shared with Mary*" False

A rough estimate of the total number of annotated instances for each approach is then given. This estimate considers a minimum of 1,000 instances to be obtained per class. Under full corpus annotation, given corpora with the same distribution as the WikiCDL corpus, 613,850 instances would have to be annotated so that all classes with at least 20 instances grow to 1,000 units. In contrast, considering the macro-average precision of 45% of automatic filtering step of set expansion, only 2,222 candidate instances would have to be annotated per class, resulting in a total of 102,222 annotated instances for all classes.

Although this estimate shows a decrease of six times in the total number of instances to be annotated, it should be added that the quality of the obtained instances should also be lower, due to the different distribution of training and testing datasets. It can thus be concluded that set expansion is more suitable for creating new training instances for classes when it would otherwise be difficult to do so with full corpus annotation. For example, the Co-Thing with Attribute (`cao`) class, illustrated by example 5.3, does not occur in the WikiCDL corpus. While a reasonable amount of effort would be needed to select texts which contain such constructions, since inspection of a large number of corpora of different genres would be required, this class is easily expanded using web queries, as indicated below:

(5.3) Relation "*Mary*" $\xrightarrow{\texttt{cao}}$ "*be*" in "*John was with Mary yesterday*"

a. "*was with **" expands to "*I was with some other soldiers*"

b. "** with Mary*" expands to "*I'm with Mary*"

For the semantic domain of modality, the primary source of cost reduction is the ability to employ sentence alignment techniques in order to propagate sentence modal value annotation across different languages, as shown in the English-Spanish example below, reducing the total cost by a factor of L, the number of languages.

(5.4) a. The directive that the Commission has put forward deals with the security for the contributors [...]. These are significant issues and as Mr Huhne knows full well, the sums involved are considerable. *In conclusion, the Commission now finds itself caught in a crossfire between Mr Kuckelkorn and Mr Huhne and therefore perhaps the golden mean is the right place for the Commission to be.*

b. La directiva que ha presentado la Comisión se ocupa de la seguridad para los contribuyentes [...]. Estos son asuntos importantes y como bien sabe el Sr. Huhne, las cifras involucradas son considerables. *En resumen, la Comisión se encuentra ahora en el fuego cruzado entre el Sr. Kuckelkorn y el Sr. Huhne y por ello quizá el punto medio sea el mejor sitio donde pueda quedarse la Comisión.*

Moreover, the annotation task is redesigned. Regular full corpus annotation requires cue expression annotation, while cue selection optimization requires only sentence modal value annotation. For an estimate of cost difference, the cue annotation in example 5.5 requires six binary decisions, compared to one binary decision for the sentence modal value annotation in example 5.6. In other words, cue annotation takes overall W_S more decisions, where W_S is the average number of expressions per sentence in the corpus.

(5.5) Many people believe X is better. Cues = { "*many people*" }

(5.6) Many people believe X is better. Uncertainty = 1

As for quality, this redesign does not necessarily imply worse performance by the interlingual analysis task, which although counter-intuitive at first, was experimentally

shown in section 3.3.3 for the Wikipedia corpus of the CoNLL-2010 Shared Task. This is particularly true for ambiguous expressions which are not easily disambiguated with currently available linguistic features. Observe example 5.7 below:

(5.7) a. Many people believe X is better. Uncertainty = 1

b. Many people believe in UFOs. Uncertainty = 0

The expression "*many people*" in sentence 5.7a indicates a vague source of information for the statement "*X is better*", whereas it is a quantifier in sentence 5.7b. When analyzing common features as presented in section 1.1.2, it becomes clear that they are not able to model disambiguation of this expression when concerning modalization. In this case, cue identification performance depends largely on whether evaluating the ambiguous expression "*many people*" as a cue or as a non-cue would produce better F-value. This was more reliably done using the cue selection approach, which explicitly considers the impact of selection of "*many people*" at a corpus-level, optimizing directly upon it.

Finally, for the semantic domain of tense, the number of annotated instances for full corpus annotation was 13,259 events, whereas for automatic inference it was 186 instances of BTSs and 169 ambiguous cases (355 instances overall), a reduction of over 37 times in the number of instances. This is illustrated in example 5.8 for the former and in example 5.9 for the latter:

(5.8) Annotation of all occurrences of present perfect with adverb "*never*" (N_O annotations for N_O occurrences):

a. I have never gone there. `E-S,R`

b. I have never traveled abroad. `E-S,R`

c. He has never been to that place. `E-S,R`

d. They have never tried sushi. `E-S,R`

(5.9) Centralized annotation of temporal senses (N_S annotations for N_S temporal senses):

a. Present perfect $(t_0 < 0, t_1 = 0, s = n)$

b. Adv. modifier past-shift (ex.: "*never*") $(t_0 < 0, t_1 = 0, s = n)$, $(t_0 =?, t_1 =?, s = p)$, etc.

Furthermore, the proposed descriptors of inspection and shifting are arguably more intuitive to annotate than Reichenbach's markers, because they do not possess an extra-linguistic component, among other reasons. While it is clear that the inspection interval

for "*never*" in "*I have never gone there*" is from negative infinity to now, the same is not true for Reichenbach's markers, for which `R` is considered to be in the present (`S,R`) in the same sentence. Another example is given in 5.10. In this sentence, an annotator may evaluate the reference point as either `S,R` or `S-R` in order to determine the position of Reichenbach's markers, whereas it is intuitive that the annotator assesses $t = 0$ when determining the inspected interval of the event "*go*", since the event may happen at subsequent moments after "now".

(5.10) a. I must go. `S,R-E` or `S-R,E`?

b. I must go. $(t_0 = 0, t_1 > 0, s = n)$

Since the correctly inferred instances have equivalent annotation to full corpus annotation using Reichenbach's markers, the quality trade-off in this approach is the number of errors, which was 8 from a total of 13,259 events (error rate of 0.06%). This is a promising result as a proof of concept on a corpus composed of texts of different genres, although further investigation in other corpora and languages would be desirable.

5.2 Contributions

This work positions itself as arguing that Interlingual Semantic Computing, despite not being suitable for the open-domain, may improve the quality of service of shallow linguistic approaches by providing sources of auxiliary semantic information. Given the Principle of Compositionality, it is possible to break the meaning of a proposition or text into smaller semantic fragments, and use these fragments for that purpose. This motivates the study of semantic analysis on a per-domain basis.

The proposed research thus studies methods for decreasing the cost of manual annotation for the selected domains of contextual semantic relations, modality and tense. A summary of the list of contributions by this research is given in this section. For Semantic Computing and NLP in general, the contributions are:

- Discussion on the exploration of linguistic domain-specificity, following the Principle of Compositionality, as a means of addressing the problem of definition of a semantic structure for the open-domain in interlingual analysis
- Proposal of methods for decreasing the cost of interlingual analysis, as a means of addressing the cost problem in interlingual analysis using supervised learning, by:
  - Identifying strategies for decreasing manual annotation cost

  - Proposing methods for the selected domains of contextual relations, modality and tense, which are all related to intra-event semantics
  - Exploring linguistic properties of these selected domains, unlike current trends of research – in NLP, generic approaches are favored over domain-specific ones, and in Linguistics, the proposal of new theoretical models is favored over simplifying semantic descriptors
  - Using annotation complexity, in order to introduce a fairer method of comparison of annotation efforts

For each domain, knowledge representation theory and existing analysis tasks were surveyed. Since tense did not have an analysis effort, one was proposed. The list of contributions for tense analysis is as follows:

- Manual annotation of a corpus with Reichenbach's markers and with the clause tree structure
- Quantitative analysis of the corpus according to SOT
- Performing clause anchoring by using a modified shift-reduce algorithm
- Modeling adverbial modification by using patterns based on temporal expressions as linguistic features
- Modeling SOT with sequential classification
- Evaluation of the proposed method, analyzing the effects with or without parent clause reference, comparing different types of clause trees, and comparing against semantic nuance perception by a shallow statistical approach

Alternative methods for decreasing costs were then proposed for each domain. For the domain of contextual semantic relations, the contributions are as follows:

- Proposal of a set expansion method which:
  - Increases the size of the training dataset while minimizing effort, instead of minimizing the size of the dataset, since a small dataset might not contain desirable word sense and word behavior features
  - Deals with class imbalance by adding web search and candidate extraction steps, unlike previous work of active learning for contextual relations

  - Adapts bootstrapped set expansion to a domain with complex structure, unlike the previous work for uncontextual lexical relations, by using oracle querying
  - Proposes a weak classifier, which focuses on positive instance classification
  - Uses feature distance metrics, which addresses classification under incomplete feature space, providing values for features which would otherwise be unknown to the classifier at the expense of loss of expressibility
  - Uses multiview distance matrices, which provides a unified view for different feature distance metrics

- Evaluation of the analysis task using expanded training sets, which indicated that new and recombined features had a positive impact on the classification task

- Decrease in annotation time complexity: Decreases from multiclass complexity (decision of one correct class among all classes) to binary complexity (decision of positive class membership), with trade-off on restriction on syntactic patterns and on the ability of the web search to return instances from the given relation class

- Better usage of resources: Although it is not guaranteed that the number of annotated instances is necessarily lower, the proposed method is able to focus annotation efforts on infrequent or selected classes, unlike full corpus annotation

For the domain of modality, the contributions are as follows:

- Removal of the dependency on transfer-based semantic annotation in cue expressions in favor of the interlingual annotation in sentence-level modality value, unlike all previous works

- Decoupling of cue selection from disambiguation, allowing independent investigation of each step

- Proposal of cue selection optimization algorithm which:
  - Focuses on the correlation between cue expressions and sentence modalization, instead of linguistic features that are not able to fully model modality
  - Runs in subquadratic time
  - Outperforms state-of-the-art systems in the general case
  - Does not require resource-intensive disambiguation, obtaining satisfactory result with lemmatization only

- Decrease in annotation time: Decreases from word-by-word cue expression annotation to sentence-by-sentence modal value binary annotation, with potential trade-off observable in modality analysis performance (varying among corpora and annotation schema)
- Decrease in annotator cost: Decreases by not requiring cue expression annotation, which is arguably simpler
- Decrease in overall annotation cost: By using sentence alignment techniques, it is possible to propagate sentence modal value to other languages

Finally, for the domain of tense, the contributions are as follows:

- Manual annotation of a corpus with BTSs and connectors
- Proposal of a tense semantics theory which:
  - Supports inference, unlike existing frameworks which assume tense semantic structure is already obtained
  - Uses more intuitive descriptors in inspection, focusing and shifting, unlike existing frameworks which use extra-linguistic descriptors
  - Investigates tense semantic contribution for all temporal entities, such as verbs, adverbial modifiers and SOT modifiers
- Evaluation of a proof of concept for automatic tense inference, which validates theory, indicating points of improvement in tense theory and in linguistic feature extraction for the task
- Decrease in annotation dataset size: Decreases from full corpus annotation to BTS annotation and ambiguous cases only, with trade-off on the number of error cases
- Decrease in annotator cost: Decreases by using a more intuitive description system which does not require as specialized annotators and is less prone to inconsistencies due to centralized annotation

5.3 Future Works

The proposed methods were presented as proofs of concept of the potential and applicability of these full corpus annotation alternatives. They may, nevertheless, be improved upon in areas such as the kernel for multiview distance matrices (a linear one is currently being used), the filtering method for cue expressions (a simple one based on a frequency

threshold is currently being used), the cue expression disambiguation technique (lemmas are currently being used), among others.

The investigation of the aspect domain, as well as its integrations with tense and modality, would also greatly enrich this work. It has been observed, for example, that tense is highly dependent on stative aspect and realis/irrealis modality. The analysis of aspect is similar to tense, since they share many linguistic features, such as verb conjugation and temporal expressions. They differ on the fact that aspect is not modeled sequentially, and that it requires better lexical features, ontological features, word sense disambiguation and frame semantics, since it is highly dependent on word behavior. In the following example, depending on the context, "*cool*" may represent an *activity* (durative event whose beginning is implied) as in sentence 5.11a or an *accomplishment* (durative event whose ending is implied) as in sentence 5.11b [Vendler, 1957].

(5.11) a. The soup *cooled* for an hour. Activity

b. The soup *cooled* in an hour. Accomplishment

Appendix A

Part-of-Speech Tags

List of Tags

The following list of part-of-speech tags was extracted from Santorini [1990], which states the 36 tags for the Penn Treebank project [Marcus et al., 1993]. The parts-of-speech encoded in the project, along with their corresponding abbreviations (tags), are reproduced herein.

- CC: Coordinating conjunction
- CD: Cardinal number
- DT: Determiner
- EX: Existential there
- FW: Foreign word
- IN: Preposition or subordinating conjunction
- JJ: Adjective
- JJR: Adjective, comparative
- JJS: Adjective, superlative
- LS: List item marker
- MD: Modal
- NN: Noun, singular or mass
- NNS: Noun, plural

- NNP: Proper noun, singular
- NNPS: Proper noun, plural
- PDT: Predeterminer
- POS: Possessive ending
- PRP: Personal pronoun
- PRP$: Possessive pronoun
- RB: Adverb
- RBR: Adverb, comparative
- RBS: Adverb, superlative
- RP: Particle
- SYM: Symbol
- TO: To
- UH: Interjection
- VB: Verb, base form
- VBD: Verb, past tense
- VBG: Verb, gerund or present participle
- VBN: Verb, past participle
- VBP: Verb, non-3rd person singular present
- VBZ: Verb, 3rd person singular present
- WDT: Wh-determiner
- WP: Wh-pronoun
- WP$: Possessive wh-pronoun
- WRB: Wh-adverb

Feature Distances

The following table details features distances for the POS tag feature type, stating the pre-defined distance values among tags.

TABLE A.1: Feature distance values for POS tags

	CC	IN	JJ	JJR	JJS	NN	NNS	NNP	NNPS	PRP	RBR	RBS	RP	TO	VB	VBD	VBG	VBN	VBP	VBZ	WDT	WP	WRB
CC	0.0	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
IN		0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
JJ			0.0	0.3	0.3	1.0	1.0	1.0	1.0	1.0	0.8	0.9	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
JJR				0.0	0.3	1.0	1.0	1.0	1.0	1.0	0.9	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
JJS					0.0	1.0	1.0	1.0	1.0	1.0	0.9	0.9	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
NN						0.0	0.1	0.1	0.2	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.7	0.7
NNS							0.0	0.2	0.1	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.7	0.7
NNP								0.0	0.1	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.7	0.7
NNPS									0.0	0.7	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.7	0.7
PRP										0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.7	0.7
RB											0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
RBR												0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
RBS													0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
TO														0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
VB															0.0	0.2	0.2	0.2	0.1	0.1	1.0	1.0	1.0
VBD																0.0	0.2	0.2	0.2	0.2	1.0	1.0	1.0
VBG																	0.0	0.2	0.2	0.2	1.0	1.0	1.0
VBN																		0.0	0.2	0.2	1.0	1.0	1.0
VBP																			0.0	0.1	1.0	1.0	1.0
VBZ																				0.0	1.0	1.0	1.0
WDT																					0.0	0.2	0.2
WP																						0.0	0.2
WRB																							0.0

Appendix B

Phrase and Dependency Relations

Phrase Structure Relations

The following lists of clause and phrase level labels were extracted from Bies et al. [1995], which states labels for the Penn Treebank project [Marcus et al., 1993]. Short explanations for each label are reproduced herein.

Clause Level

- S: Simple declarative clause
- SBAR: Clause introduced by a subordinating conjunction
- SBARQ: Direct question introduced by a wh-word or a wh-phrase
- SINV: Inverted declarative sentence
- SQ: Inverted yes/no question, or main clause of a wh-question, following the wh-phrase in SBARQ

Phrase Level

- ADJP: Adjective phrase
- ADVP: Adverb phrase
- CONJP: Conjunction phrase
- FRAG: Fragment
- INTJ: Interjection

- LST: List marker
- NAC: Not a constituent
- NP: Noun phrase
- NX: Used within certain complex NPs to mark the head of the NP
- PP: Prepositional phrase
- PRN: Parenthetical
- PRT: Particle
- QP: Quantifier phrase
- RRC: Reduced relative clause
- UCP: Unlike coordinated phrase
- VP: Verb phrase
- WHADJP: Wh-adjective phrase
- WHAVP: Wh-adverb phrase
- WHNP: Wh-noun phrase
- WHPP: Wh-prepositional phrase
- X: Unknown, uncertain, or unbracketable

Dependency Relations

The following list of 52 dependencies was extracted from De Marneffe & Manning [2008]. The dependencies, along with a brief description, are reproduced herein.

- abbrev: Abbreviation modifier
- acomp: Adjectival complement
- advcl: Adverbial clause modifier
- advmod: Adverbial modifier
- agent: Agent
- amod: Adjectival modifier

- appos: Appositional modifier
- attr: Attributive
- aux: Auxiliary
- auxpass: Passive auxiliary
- cc: Coordination
- ccomp: Clausal complement
- complm: Complementizer
- conj: Conjunct
- cop: Copula
- csubj: Clausal subject
- csubjpass: Clausal passive subject
- dep: Dependent
- det: Determiner
- dobj: Direct object
- expl: Expletive
- infmod: Infinitival modifier
- iobj: Indirect object
- mark: Marker
- mwe: Multi-word expression
- neg: Negation modifier
- nn: Noun compound modifier
- npadvmod: Noun phrase as adverbial modifier
- nsubj: Nominal subject
- nsubjpass: Passive nominal subject
- num: Numeric modifier
- number: Element of compound number

- parataxis: Parataxis
- partmod: Participial modifier
- pcomp: Prepositional complement
- pobj: Object of a preposition
- poss: Possession modifier
- possessive: Possessive modifier
- preconj: Preconjunct
- predet: Predeterminer
- prep: Prepositional modifier
- prepc: Prepositional clausal modifier
- prt: Phrasal verb particle
- punct: Punctuation
- purpcl: Purpose clause modifier
- quantmod: Quantifier phrase modifier
- rcmod: Relative clause modifier
- ref: Referent
- rel: Relative
- tmod: Temporal modifier
- xcomp: Open clausal complement
- xsubj: Controlling subject

Appendix C

Discourse Relations

Rhetorical Relations

The following list of rhetorical relations was extracted from Carlson et al. [2001]. The 16 relation superclasses and 78 classes are reproduced herein.

- Attribution: attribution, attribution-negative
- Background: background, circumstance
- Cause: cause, result, consequence
- Comparison: comparison, preference, analogy, proportion
- Condition: condition, hypothetical, contingency, otherwise
- Contrast: contrast, concession, antithesis
- Elaboration: elaboration-additional, elaboration-general-specific, elaboration-part-whole, elaboration-process-step, elaboration-object-attribute, elaboration-set-member, example, definition
- Enablement: purpose, enablement
- Evaluation: evaluation, interpretation, conclusion, comment
- Explanation: evidence, explanation-argumentative, reason
- Joint: list, disjunction
- Manner-Means: manner, means

- Topic-Comment: problem-solution, question-answer, statement-response, topic-comment, comment-topic, rhetorical-question
- Summary: summary, restatement
- Temporal: temporal-before, temporal-after, temporal-same-time, sequence, invertedsequence
- Topic Change: topic-shift, topic-drift

Penn Discourse Treebank Relations

The following list of discourse relations was extracted from Prasad et al. [2008]. Relation classes, types and subtypes are reproduced herein.

Temporal Relations

The situations described in `Arg1` and `Arg2` are temporally related.

- Asynchronous: Precedence, succession
- Synchronous

Contingency Relations

The situations described in `Arg1` and `Arg2` are causally influenced.

- Cause: Reason, result
- Pragmatic cause
- Condition: Hypothetical, general, factual present, factual past, unreal present, unreal past
- Pragmatic condition: Relevance, implicit assertion

Comparison Relations

The situations described in `Arg1` and `Arg2` are compared and differences between them are identified.

- Contrast: Juxtaposition, opposition
- Pragmatic contrast
- Concession: Expectation, contra-expectation

Expansion Relations

The situation described in `Arg2` provide information deemed relevant to the situation described in `Arg1`.

- Instantiation
- Restatement: Specification, equivalence, generalization
- Alternative: Conjunctive, disjunctive, chosen alternative
- Exception
- Conjunction
- List

Appendix D

CDL Relation Classes

The following list of CDL.nl relation classes was extracted from Uchida et al. [2005]. The class label and a brief description are reproduced herein. Notice that the macro-grouping is a modification of the original available in Zhu et al. [2005].

Agent Relations

Relations that indicate entities that initiate actions or that are in a certain state.

- Agent (`agt`): Indicates a thing in focus that initiates an action
- Thing with attribute (`aoj`): Indicates a thing that is in s state or has an attribute
- Co-agent (`cag`): Indicates a thing not in focus that initiates an implicit event that is done in parallel
- Co-thing with attribute (`cao`): Indicates a thing not in focus that is in a parallel state
- Partner (`ptn`): Indicates an indispensable non-focused initiator of an action

Object Relations

Relations that indicate entities that are affected by actions or states.

- Affected thing (`obj`): Indicates a thing in focus that is directly affected by an event or state

- Affected co-thing (`cob`): Indicates a thing that is directly affected by an implicit event done in parallel or an implicit state in parallel
- Affected place (`opl`): Indicates a place in focus affected by an event
- Beneficiary (`ben`): Indicates an indirectly related beneficiary or victim of an event or state

Place Relations

Relations that indicate places that an action occurs or a state is true.

- Place (`plc`): Indicates a place where an event occurs, or a state that is true, or a thing that exists
- Initial place (`plf`): Indicates a place where an event begins or a state that becomes true
- Final place (`plt`): Indicates a place where an event ends or a state that becomes false
- Scene (`scn`): Indicates a scene where an event occurs, or state is true, or a thing exists

Instrument Relations

Relations that indicate instruments or means to carry out an event.

- Instrument (`ins`): Indicates an instrument to carry out an event
- Method or means (`met`): Indicates means to carry out an event

State Relations

Relations that indicate state of an event.

- Source or initial state (`src`): Indicates the initial state of an object or thing initially associated with the object of an event

- Goal or final state (`gol`): Indicates a final state of object or a thing finally associated with the object of an event
- Intermediate place or state (`via`): Indicates an intermediate place or state of an event

Time Relations

Relations that indicate the time that an action occurs or a state is true.

- Time (`tim`): Indicates the time an event occurs or a state is true
- Initial time (`tmf`): Indicates the time an event starts or a state becomes true
- Final time (`tmt`): Indicates a time an event ends or a state becomes false
- Duration (`dur`): Indicates a period of time during which an event occurs or a state exists

Manner Relations

Relations that indicate the way that an event is conducted, or basis for comparison.

- Manner (`man`): Indicates a way to carry out an event or the characteristics of a state
- Basis (`bas`): Indicates a thing used as the basis (standard) of comparison

Logical Relations

Relations that indicate logical connection between entities or events.

- Conjunction (`and`): Indicates a partner to have conjunctive relation to
- Intersection (`int`): Indicates all common instances to have with a partner concept
- Disjunction (`or`): Indicates a partner to have disjunctive relation to

Concept Relations

Relations that indicate conceptual connection between two entities.

- Equivalence (`equ`): Indicates an equivalent concept
- Included or kind of (`icl`): Indicates an upper concept or a more general concept
- Instance of (`iof`): Indicates a class concept that an instance belongs to

Cause Relations

Relations that indicate condition, purpose or reason of an event.

- Condition (`con`): Indicates a non-focused event or state that conditions a focused event or state
- Purpose (`pur`): Indicates the purpose or objective of an agent of an event or the purpose of a thing that exists
- Reason (`rsn`): Indicates a reason why an event or a state happens

Sequence Relations

Relations that indicate events that are in a cronological sequence.

- Co-occurrence (`coo`): Indicates a co-occurrent event or state for a focused event or state
- Sequence (`seq`): Indicates a prior event or state of a focused event or state

Restrictive Relations

Relations that indicate restrictive characteristics of an entity.

- Content (`cnt`): Indicates the content of a concept
- Range (from-to) (`fmt`): Indicates a range between two things

- Origin (`fmr`): Indicates an initial state of a thing or a thing initially associated with the focused thing
- Modification (`mod`): Indicates a thing that restricts a focused thing
- Name (`nam`): Indicates the name of a thing
- Proportion, rate or distribution (`per`): Indicates a basis or unit of proportion, rate or distribution
- Part of (`pof`): Indicate a concept of which a focused thing is a part
- Possession (`pos`): Indicates the possessor of a thing
- Quantity (`qua`): Indicates the quantity of a thing or unit
- Sentence head (`shd`): Indicates a number, a mark or a thing that shows the position of a sentence, a paragraph or a chapter in a document or a book
- Destination (`to`): Indicates a final state of a thing or a final thing (destination) associated with the focused thing

List of Publications

Horie, A. K. & Ishizuka, M. (2012). Set Expansion of Contextual Semantic Relations: An alternative for full corpus annotation for supervised classification. *International Journal of Semantic Computing*, 6(01), 93–109.

Horie, A. K., Tanaka-Ishii, K., & Ishizuka, M. (2012). Verb Temporality Analysis using Reichenbach's Tense System. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)* (pp. 471–482). Mumbai, India: Association for Computational Linguistics.

Horie, A. K. & Tanaka-Ishii, K. (2014a). Sentence Hedge Detection without Cue Annotation: A heuristic cue selection approach. *Journal of Natural Language Processing*, 21(01), 27–40.

Horie, A. K. & Tanaka-Ishii, K. (2014b). Composite Temporal Structure: For the automation of inference of tense semantics. *Language Resources and Evaluation*. Under review.

Bibliography

Agichtein, E., Gravano, L., Pavel, J., Sokolova, V., & Voskoboynik, A. (2001). Snowball: A Prototype System for Extracting Relations from Large Text Collections. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 612).

Arora, S., Nyberg, E., & Rosé, C. P. (2009). Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing* (pp. 18—-26).: Association for Computational Linguistics.

Asher, R. E. (1994). *The Encyclopedia of Language and Linguistics*. Pergamon, 1 edition.

Avramidis, E. & Koehn, P. (2008). Enriching Morphologically Poor Languages for Statistical Machine Translation. *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008): Human Language Technologies (HLT)*, (pp. 763–770).

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING 1998)* (pp. 86–90).

Baker, K., Bloodgood, M., Dorr, B. J., Filardo, N. W., Levin, L. S., & Piatko, C. D. (2010). A Modality Lexicon and its use in Automatic Tagging. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*.

Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M. A., & Schasberger, B. (1995). *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. Technical report, University of Pennsylvania.

Binnick, R. I. (1991). *Time and the Verb: A guide to tense and aspect*. Oxford University Press.

Blackburn, P. (1994). Tense, Temporal Reference, and Tense Logic. *Journal of Semantics*, 11(1-2), 83–101.

Blackburn, P. & Bos, J. (2003). Representation and Inference for Natural Language. *A First Course in Computational Semantics*.

Blum, A. & Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory* (pp. 92—-100).: ACM.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2009). Measuring the Similarity Between Implicit Semantic Relations from the Web. In *Proceedings of the 18th International Conference on World Wide Web (WWW 2009)* (pp. 651–660). New York, NY, USA: Association for Computing Machinery.

Bollegala, D., Matsuo, Y., & Ishizuka, M. (2010). Relational Duality: Unsupervised Extraction of Semantic Relations between Entities on the Web. In *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)* (pp. 151–160). New York, NY, USA: Association for Computing Machinery.

Bunescu, R. C. & Mooney, R. J. (2005). A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)* (pp. 724–731).: Association for Computational Linguistics.

Carlson, L., Marcu, D., & Okurowski, M. E. (2001). Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, volume 16 (pp. 1–10).: Association for Computational Linguistics.

Carreras, X. & Màrquez, L. (2003). Phrase Recognition by Filtering and Ranking with Perceptrons. In *Proceedings of the 2003 Recent Advances in Natural Language Processing (RANLP-2003)* (pp. 205–216).

Carreras, X. & Màrquez, L. (2004). Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL 2004)*: Association for Computational Linguistics.

Carreras, X. & Màrquez, L. (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005)*: Association for Computational Linguistics.

Chakravarthy, V., Joshi, S., Ramakrishnan, G., Godbole, S., & Balakrishnan, S. (2008). Learning Decision Lists with Known Rules for Text Mining. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing* (pp. 835–840).

Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings of the 2000 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)* (pp. 132–139).: Association for Computational Linguistics.

Chieu, H. L. & Ng, H. T. (2003). Named Entity Recognition with a Maximum Entropy Approach. In *Proceedings of the 7th Conference on Computational Natural Language Learning (CoNLL 2003)*.

Chomsky, N. (1957). *Syntactic Structures*. The Hague: Mouton.

Collins, M. (2003). Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29(4), 589–637.

Comrie, B. (1985). *Tense*. Cambridge University Press.

Dagan, I. & Engelson, S. P. (1995). Committee-based Sampling for Training Probabilistic Classifiers. In *International Conference on Machine Learning*, volume 95 (pp. 150—-157).

Dash, M. & Liu, H. (1997). Feature Selection for Classification. *Intelligent Data Analysis*, 1(3), 131–156.

De Marneffe, M. C. & Manning, C. D. (2008). *Stanford Typed Dependencies Manual*. Technical report, Stanford University.

de Saussure, F. (1916). *Course in General Linguistics*. Fontana/Collins, 3rd edition.

Declerck, R. (1986). From Reichenbach (1947) to Comrie (1985) and beyond: towards a theory of tense. *Lingua*, 70(4), 305–364.

Derczynski, L. & Gaizauskas, R. (2011). An Annotation Scheme for Reichenbach's Verbal Tense Structure. In *Workshop on Interoperable Semantic Annotation* (pp.1̃0).

Dorr, B. J., Jordan, P. W., & Benoit, J. W. (1999). A Survey of Current Paradigms in Machine Translation. *Advances in Computers*, 49, 1–68.

Dowty, D. R. (1979). *Word Meaning and Montague Grammar: The semantics of verbs and times in generative semantics and in Montague's PTQ*, volume 7. Springer.

Engelson, S. P. & Dagan, I. (1996). Minimizing Manual Annotation Cost in Supervised Training from Corpora. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* (pp. 319—-326).: Association for Computational Linguistics.

Erdmann, M., Maedche, A., Schnurr, H.-P., & Staab, S. (2000). From Manual to Semi-Automatic Semantic Annotation: About ontology-based text annotation tools. In *COLING-2000 Workshop on Semantic Annotation and Intelligent Content* (pp. 79—-85).: Association for Computational Linguistics.

Farkas, R., Vincze, V., Móra, G., Csirik, J., & Szarvas, G. (2010). The CoNLL-2010 Shared Task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task* (pp. 1–12).: Association for Computational Linguistics.

Fellbaum, C., Ed. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA; London: MIT Press.

Fillmore, C. J. (1967). The case for case.

Fillmore, C. J. & Baker, C. F. (2001). Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*.

Fleuret, F. (2004). Fast Binary Feature Selection with Conditional Mutual Information. *The Journal of Machine Learning Research*, 5, 1531–1555.

Frege, G. (1892). On Concept and Object. In M. Black & P. Geach (Eds.), *In Translations from the Philosophical Writings of Gottlob Frege, 1952.* (pp. 56–78).: Oxford: Blackwell.

Georgescul, M. (2010). A Hedgehop over a Max-Margin Framework using Hedge Cues. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task* (pp. 26–31).: Association for Computational Linguistics.

Giménez, J. & Màrquez, L. (2003). Fast and Accurate Part-of-Speech Tagging: The SVM approach revisited. *Proceedings of the 2003 Recent Advances in Natural Language Processing (RANLP-2003).*

Goodman, J. (2002). An Incremental Decision List Learner. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* (pp. 17–24).: Association for Computational Linguistics.

Guéron, J. (2007). On tense and aspect. *Lingua*, 117(2), 367–391.

He, S. & Gildea, D. (2006). *Self-Training and Co-Training for Semantic Role Labeling: Primary report.* Technical report, University of Rochester.

Hernault, H., Bollegala, D., & Ishizuka, M. (2010). A Semi-Supervised Approach to Improve Classification of Infrequent Discourse Relations using Feature Vector Extension. In *Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing (EMNLP 2010)* Massachusetts, USA.

Hopper, P. J. (1979). Aspect and Foregrounding in Discourse. *Discourse and Syntax*.

Hornstein, N. (1990). *As Time Goes By: Tense and universal grammar*. MIT Press Cambridge.

Japkowicz, N. & Stephen, S. (2002). The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis*, 6(5), 429–449.

Joachims, T. (1998). *Text categorization with Support Vector Machines: Learning with many relevant features*. Springer.

Kannan, R., Vempala, S., & Vetta, A. (2004). *On Clusterings: Good, bad and spectral*. Technical Report 3, Yale University.

Khoo, C. S. G. & Na, J. C. (2007). Semantic Relations in Information Science. *Annual Review Information Science & Technology*, 40(1), 157–228.

Koehn, P. (2010). *Statistical Machine Translation*. New York, NY, USA: Cambridge University Press, 1 edition.

Kolomiyets, O., Bethard, S., & Moens, M. F. (2012). Extracting narrative timelines as temporal dependency structures. In *Proceedings of 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*: ACL.

Kučera, H. & Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Dartmouth Publishing Group.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)* (pp. 282–289).: Morgan Kaufmann Publishers.

Leech, G. (1983). *Principles of Pragmatics*. Longman London.

Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. University of Chicago Press.

Li, H., Bollegala, D., Matsuo, Y., & Ishizuka, M. (2011). Using Graph Based Method to Improve Bootstrapping Relation Extraction. In *Proceedings of the 2011 Conferences on Computational Linguistics and Natural Language Processing (CICLing 2011)* Tokyo, Japan.

Light, M., Qiu, X. Y., & Srinivasan, P. (2004). The Language of Bioscience: Facts, speculations, and statements in between. In *Proceedings of BioLink 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for users* (pp. 17–24).

Lin, H.-T., Lin, C.-J., & Weng, R. C. (2007). A Note on Platt's Probabilistic Outputs for Support Vector Machines. *Machine Learning*, 68(3), 267–276.

Lunn, P. V. (1995). The evaluative function of the Spanish subjunctive. *Modality in Grammar and Discourse*, 32, 429–449.

Maekawa, K. (2007). Kotonoha and BCCWJ: Development of a balanced corpus of contemporary written Japanese. In *Corpora and Language Research: Proceedings of the 1st International Conference on Korean Language, Literature, and Culture* (pp. 158–177).

Mani, I. & Wilson, G. (2000). Robust Temporal Processing of News. In *Proceedings of 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)* (pp. 69–76).: ACL.

Mann, W. C. & Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3), 243–281.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA, USA: MIT Press.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313–330.

Matsuyoshi, S., Eguchi, M., Sao, C., Murakami, K., Inui, K., & Matsumoto, Y. (2010). Annotating Event Mentions in Text with Modality, Focus, and Source Information. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*.

Mazur, P. & Dale, R. (2010). WikiWars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing (EMNLP 2010)* (pp. 913–922).: Association for Computational Linguistics.

McShane, M., Nirenburg, S., & Zacharski, R. (2004). Mood and Modality: Out of theory and into the fray. *Natural Language Engineering*, 10(1), 57–89.

Medlock, B. & Briscoe, T. (2007). Weakly Supervised Learning for Hedge Classification in Scientific Literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)* (pp. 992–999). Prague, Czech Republic: Association for Computational Linguistics.

Miltsakaki, E., Prasad, R., Joshi, A. K., & Webber, B. L. (2004). The Penn Discourse Treebank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*: Citeseer.

Moore, R. C. (2002). Fast and Accurate Sentence Alignment of Bilingual Corpora. In *Machine Translation: From Research to Real Users* (pp. 135—-144). Springer.

Morante, R., Van Asch, V., & Daelemans, W. (2010). Memory-based resolution of in-sentence scopes of hedge cues. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task* (pp. 40–47).: Association for Computational Linguistics.

Moulin, B. & Dumas, S. (1994). The Temporal Structure of a Discourse and Verb Tense Determination. *Conceptual Structures: Current Practices*, (pp. 45–68).

Musan, R. (2001). The Present Perfect in German: Outline of its semantic composition. *Natural Language and Linguistic Theory*, 19(2), 355–401.

Navigli, R. & Lapata, M. (2010). An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4), 678–692.

Navigli, R. & Ponzetto, S. P. (2012). Multilingual WSD with just a few lines of code: The BabelNet API. In *Proceedings of the ACL 2012 System Demonstrations* (pp. 67–72).: Association for Computational Linguistics.

Nirenburg, S., Beale, S., & McShane, M. (2008). Baseline evaluation of WSD and semantic dependency in OntoSem. In *Proceedings of the 2008 Conference on Semantics in Text Processing* (pp. 315–326).: Association for Computational Linguistics.

Nordlinger, R. & Sadler, L. (2004). Tense Beyond the Verb: Encoding clausal tense/aspect/modality on nominal dependents. *Natural Language and Linguistic Theory*, 22(3), 597–641.

Okazaki, N. (2007). CRFsuite: A fast implementation of conditional random fields (CRFs).

Palmer, F. R. (2001). *Mood and Modality*. Cambridge.

Palmer, M. (2009). SemLink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference* (pp. 9–15).

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1), 71–106.

Pedersen, T. & Kolhatkar, V. (2009). WordNet::SenseRelate::AllWords: A broad coverage word sense tagger that maximizes semantic relatedness. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session* (pp. 17–20).: Association for Computational Linguistics.

Platt, J. C. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers* (pp. 61–74).

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., & Webber, B. L. (2008). The Penn Discourse TreeBank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*: Citeseer.

Prior, A. N. (1967). *Past, Present and Future*, volume 154. Clarendon Press Oxford.

Pustejovsky, J., Castano, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G., & Radev, D. (2003a). TimeML: Robust specification of event and temporal expressions in text. *New Directions in Question Answering*, 2003, 28–34.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., & Others (2003b). The Timebank corpus. In *Corpus Linguistics*, volume 2003 (pp.40).

Reichenbach, H. (1947). *Elements of Symbolic Logic*.

Ringger, E. K., Carmen, M., Haertel, R., Seppi, K. D., Lonsdale, D., McClanahan, P., Carroll, J. L., & Ellison, N. (2008). Assessing the Costs of Machine-Assisted Corpus Annotation through a User Study. In *Language Resources and Evaluation Conference*, volume 8 (pp. 3318—-3324).

Roth, D. & Small, K. (2006). Margin-based Active Learning for Structured Output Spaces. In *Proceedings of the 17th European Conference on Machine Learning* (pp. 413—-424).: Springer.

Santorini, B. (1990). Part-of-speech Tagging Guidelines for the Penn Treebank Project (3rd Revision).

Saurí, R., Knippen, R., Verhagen, M., & Pustejovsky, J. (2005). Evita: A robust event recognizer for QA systems. In *Proceedings of the 2005 Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)* (pp. 700–707).: Association for Computational Linguistics.

Saurí, R. & Pustejovsky, J. (2009). FactBank: A corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3), 227–268.

Schuler, K. K. (2005). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon.* PhD thesis, University of Pennsylvania.

Settles, B. (2009). *Active Learning Literature Survey.* Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Settles, B., Craven, M., & Friedland, L. (2008). Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning* (pp. 1—-10).

Shi, J. & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888–905.

Smith, C. S. (1978). The syntax and interpretation of temporal expressions in English. *Linguistics and Philosophy*, 2(1), 43–99.

Szarvas, G. (2008). Hedge Classification in Biomedical Texts with a Weakly Supervised Selection of Keywords. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008)*: Association for Computational Linguistics.

Tapanainen, P. & Järvinen, T. (1997). A Non-Projective Dependency Parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing* (pp. 64–71).: Association for Computational Linguistics.

Tjong, E. F., Sang, K., & Déjean, H. (2001). Introduction to the CoNLL-2001 Shared Task: Clause identification. In *Proceedings of ACL-2001 Workshop on Computational Natural Language Learning (CoNLL 2001)* (pp.$\tilde{8}$).: ACL.

Tomlin, R. S. (1985). Foreground-Background Information and the Syntax of Subordination. *Text-Interdisciplinary Journal for the Study of Discourse*, 5(1-2), 85–122.

Trask, R. L. (1993). *A Dictionary of Grammatical Terms in Linguistics.* Psychology Press.

Uchida, H. & Zhu, M. (2002). Universal Word and UNL Knowledge Base. In *International Conference on Universal Knowledge and Language (ICUKL)* Goa, India.

Uchida, H., Zhu, M., & Senta, T. D. (2005). *Universal Networking Language.* UNDL Foundation.

Vauquois, B. (1968). A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation. In *IFIP Congress*, volume 68 (pp. 254–260).

Vendler, Z. (1957). Verbs and Times. *The Philosophical Review*, (pp. 143–160).

Verhagen, M., Mani, I., Saurí, R., Knippen, R., Jang, S. B., Littman, J., Rumshisky, A., Phillips, J., & Pustejovsky, J. (2005). Automating temporal annotation with TARSQI. In *Proceedings of ACL-2005 on Interactive Poster and Demonstration Sessions* (pp. 81–84).: ACL.

Vincze, V., Szarvas, G., Farkas, R., Móra, G., & Csirik, J. (2008). The BioScope Corpus: Biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, 9(Suppl 11), S9.

Von Fintel, K. (2006). Modality and Language. *Encyclopedia of Philosophy.*

Webber, B. L. & Joshi, A. K. (1998). Anchoring a lexicalized tree-adjoining grammar for discourse. In *Workshop on Discourse Relations and Discourse Markers, COLING 1998* (pp. 86–92).

Wenyin, L., Dumais, S., Sun, Y., Zhang, H., Czerwinski, M., & Field, B. (2001). Semi-Automatic Image Annotation. In *Interact 2001* (pp. 326—-333).

Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3), 165–210.

Wu, D. & Fung, P. (2009). Can Semantic Role Labeling Improve SMT. In *Proceedings of the 13th Annual Conference of the EAMT* (pp. 218—-225).

Yan, Y. (2010). *Relation Extraction from Web Contents with Linguistic and Web Features.* PhD thesis, University of Tokyo, Tokyo, Japan.

Yan, Y., Matsuo, Y., Ishizuka, M., & Yokoi, T. (2008). Annotating an Extension Layer of Semantic Structure for Natural Language Text. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing (ICSC 2008)* (pp. 174–181). Washington, DC, USA: IEEE Computer Society.

Yokoi, T., Yasuhara, H., Uchida, H., Zhu, M., & Hashida, K. (2005). CDL (Concept Description Language): A Common Language for Semantic Computing. In *Online Proceedings of the Workshop on the Semantic Computing Initiative (SeC 2005)* Makuhari, Japan.

Yu, L. & Liu, H. (2003). Feature Selection for High-Dimensional Data: A fast correlation-based filter solution. In *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)* (pp. 856–863).

Zhao, Q., Sun, C., Liu, B., & Cheng, Y. (2010). Learning to Detect Hedges and their Scope using CRF. In *Proceedings of the 14th Conference on Computational Natural Language Learning—Shared Task* (pp. 100–105).: Association for Computational Linguistics.

Zhu, M., Uchida, H., & Yokoi, T. (2005). *Specification of CDL.nl.* Technical report, Institute of Semantic Computing of Japan.