NATURAL LANGUAGE INFERENCE: FROM TEXTUAL ENTAILMENT TO CONVERSATION ENTAILMENT

By

Chen Zhang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science

2010

ABSTRACT

NATURAL LANGUAGE INFERENCE: FROM TEXTUAL ENTAILMENT TO CONVERSATION ENTAILMENT

By

Chen Zhang

Automatic inference from natural language is a critical yet challenging problem for many language-related applications. To improve the ability of natural language inference for computer systems, recent years have seen an increasing research effort on textual entailment. Given a piece of text and a hypothesis statement, the task of textual entailment is to predict whether the hypothesis can be inferred from the text.

The studies on textual entailment have mainly focused on automated inference from archived news articles. As more data on human-human conversations become available, it is desirable for computer systems to automatically infer information from conversations, for example, knowledge about their participants. However, unlike news articles, conversations have many unique features, such as turn-taking, grounding, unique linguistic phenomena, and conversation implicature. As a result, the techniques developed for textual entailment are potentially insufficient for making inference from conversations.

To address this problem, this thesis conducts an initial study to investigate conversation entailment: given a segment of conversation script, and a hypothesis statement, the goal is to predict whether the hypothesis can be inferred from the conversation segment. In this investigation, we first developed an approach based on dependency structures. This approach achieved 60.8% accuracy on textual entailment, based on the testing data of PASCAL RTE-3 Challenge. However, when applied to conversation entailment, it achieved an accuracy of 53.1%. To improve its performance on conversation entailment, we extended our models by incorporating additional linguistic features from conversation utterances and structural features from conversation

discourse. Our enhanced models result in a prediction accuracy of 58.7% on the testing data, significantly above the baseline performance ($p < 0.05$).

This thesis provides detailed descriptions about semantic representations, computational models, and their evaluations on conversation entailment.

ACKNOWLEDGMENT

My acknowledgments to Dr. Joyce Chai, my advisor. You lead me in the field of research for all these many years, you gave me so many advises, and you worked so much with me on this thesis.

Acknowledgments to my guidance committee, Dr. John Hale, Dr. Rong Jin, and Dr. Pang-Ning Tan. Thanks for your valuable comments and directions. They greatly helped this thesis.

To my fellow workers, Matthew Gerber, Tyler Baldwin, Zahar Prasov, Shaolin Qu, and Changsong Liu. You shared your ideas and knowledge. They are very important to this work.

To Marie Lazar, Timothy Aubel, Sarah Deighan, Jeff Winship and many others. Your contributions to the data collection and annotation are very much appreciated. They made this work possible.

To mom and dad. You are always with me.

Thank you, Jean.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

While we human, based on our linguistic and world knowledge and reasoning capabilities, are able to make inference and derive knowledge and conclusions from what we communicate to each other, automated inference from natural language has been a significant challenge for NLP systems. This is due to many reasons:

1. The variability, flexibility, and ambiguity from the language itself.

2. The representation of knowledge in computer systems and the scope of the world knowledge.

3. The capabilities that support automated reasoning.

A tremendous amount of research has been done in pursuing all the above directions. Recent efforts which have touched upon all these directions are the five events of PASCAL RTE (Recognizing Textual Entailment) Challenge [8, 10, 22, 36, 37].

The PASCAL RTE Challenge formulates natural language inference problem as a textual entailment problem. It provides a concrete, yet informal definition of the problem: a *textual entailment* is a directional relationship between pairs of text expressions, denoted by T - the entailing "Text", and H - the entailed "Hypothesis". T is said to *entail* H if we can infer H from the meaning of T. Examples of

Table 1.1: Examples of text-hypothesis pairs for textual entailment

Text	Hypothesis	Entailed
iTunes software has seen strong sales in Europe.	Strong sales for iTunes in Europe.	True
Cavern Club sessions paid the Beatles £15 evenings and £5 lunchtime.	The Beatles perform at Cavern club at lunchtime.	True
Sharon warns Arafat could be targeted for assassination.	Prime minister targeted for assassination.	False
Mitsubishi Motors Corp.'s new vehicle sales in the US fell 46 percent in June.	Mitsubishi sales rose 46 percent.	False

text-hypothesis pairs from the PASCAL RTE Challenge, together with the labels of whether H is entailed from T, are shown in Table 1.1.

Because complete, accurate, open-domain natural language understanding is far beyond current capabilities, nearly all efforts in this area have sought to extract the maximum mileage from quite limited semantic representations. There are three major classes of approaches to the textual entailment problem: the IR-based approaches, the logic-based approaches, and the graph-based approaches. An overview of these approaches can be found in Section 2.1.

Successfully recognizing textual entailment has many potential applications such as text retrieval, question answering, information extraction, document summarization, and machine translation evaluation.

While PASCAL provides a concrete platform for studying natural language inference, its particular focus is on text. The data are all from well-formed newswire articles in a monologue fashion. Nowadays, more and more conversation scripts has become available, such as call center records, conference transcripts, public speeches and interviews, court records, online chatting, and so on. They contain vast amount of information, such as profiling information of conversation participants and information about their social relations, beliefs, and opinions. Therefore, the capability to automatically infer knowledge and facts from these data has become increasingly important. One question is, can we follow the PASCAL practice and study natural

language inference from the dialogue setting?

On the one hand, although a conversation is a communication by two or more people, it is essentially a kind of information expressed by natural language, as is the case for text. Therefore, making inference from conversations requires similar techniques as textual entailment such as language modeling, lexical processing, syntactic parsing, and semantic understanding, and also shares the same tools such as reasoning and world knowledge.

On the other hand, conversations also have many unique characteristics that distinguish them from text. The key distinctive features include turn-taking, grounding, implicature, and different linguistic phenomena. They can also contain information that is unique to themselves. For example, in a task-oriented conversation, we are interested in whether the task is accomplished in the end; in a cooperative conversation, we may be interested in how well the participants cooperated with each other; and in a debate, we may want to know which party performs better or which one actually wins the debate. These tasks involve not only the processing of lexica, syntax, and semantics, but also the recognition of dialogue intention and conversation structure. Therefore the inference from conversation scripts is a more challenging task.

Thus, it is the goal of this thesis to take an initial investigation on natural language inference from conversation scripts. Inspired by textual entailment, we formulate this problem as *conversation entailment*: given a segment of conversation discourse D and a hypothesis H, the goal is to identify whether H can be entailed from D. For example, below is a short segment of conversation script, together with a list of hypotheses.

Conversation segment:

A: Um, yeah, I would like to talk about how you dress for work, and, and, um, what do you normally, what type of outfit do you normally have to wear?

B: Well, I work in, uh, Corporate Control, so we have to dress kind of nice, so I usually wear skirts and sweaters in the winter time, slacks, I guess.

Hypotheses:

1. A wants to know B's dress code at work.
2. B works in Corporate Control.
3. The speakers have to dress nice at work.

In this example, the first two hypotheses can be entailed from the conversation segment, while the third one cannot.

1.1 Research Objectives and Overview

To study the problem of conversation entailment, this thesis particularly examines the following issues:

1. To what degree the techniques developed for textual entailment can be re-used for conversation entailment?

2. What unique characteristics of conversations should be modeled and incorporated for conversation entailment?

3. How to combine linguistic, discourse, and context features together to develop an automated system for conversation entailment?

To address the above questions, in this thesis we have conducted the following work:

1. We created a database of examples on conversation entailment following the PASCAL practice of textual entailment to facilitate our research objectives. We selected 50 conversations from the Switchboard corpus [38] and had 15 volunteer annotators read the selected conversations and create hypotheses about

participants. As a result, a total of 1096 entailment examples were created. Each example consists of a conversation segment, a hypothesis statement, and a truth value indicating whether the hypothesis can be entailed from the conversation segment given the whole history of that conversation session. Inspired by previous work [34, 49], we particularly asked annotators to provide hypotheses that address the profiling information of the participants, their opinions and desires, as well as the communicative intents (e.g., agreements or disagreements) between participants.

The entailment judgement for each example was further independently annotated by four annotators (who were not the original contributors of the hypotheses). As a result, on average each entailment example (i.e., a pair of conversation segment and hypothesis) received five judgements. We removed the entailment examples that have less than 75% agreement among human annotators, and divided the remaining data into a development set of 291 examples and a test set of 584 examples.

2. We developed a probabilistic framework that facilitates the solution of both textual entailment and conversation entailment problems. This framework first represents all forms of language in terms of dependency structures, and then conducts a two-stage procedure to predict the entailment relation.

   In the first stage, the nodes in the dependency structure of the hypothesis side are aligned to the nodes in the dependency structure of the premise side (i.e., text or conversation segment). In the second stage, the relations in the dependency structure of the hypothesis are predicted to be entailed or not entailed. Probabilistic decomposition allows the system to break down the decision of whether the entire hypothesis is entailed into a series of decisions that whether each relation in the dependency structure of the hypothesis is

entailed.

We developed a baseline approach based on this framework that is driven by textual entailment, and applied it to solve the conversation entailment problem.

3. We identified unique language behaviors that distinguish conversations from text, which may have potential influence on the entailment decision. We developed a representation of conversation structure that augments the dependency structure representation. This is done by expanding the dependency structure of conversation segment, incorporating turn-taking, speaker, and dialogue act information. We show through experiments that this feature is very important in predicting conversation entailment. Combined with enhanced computational models (introduced below), the modeling of conversation structure improves the performance by an absolute difference of 4.8% on the test data. Particularly, we have found that such modeling is especially important for the inference of participants' communicative intents.

4. We developed enhanced computational models that integrates shallow semantic characterization for predicting conversation entailment. String representation is used to describe the long distance relationship between any two language constituents in a dependency structure. Such relational features in syntactic parse structures, which have been used in other language processing tasks such as semantic role labeling [74], are known as an effective way to model "shallow semantics" in language. However, their usage in entailment tasks has not yet been explored.

   We demonstrated through our experiments that the enhanced feature is an important way to characterize the (shallow) semantic relation between two language constituents. This feature helps to make the prediction of whether a certain kind of relation in the hypothesis statement is entailed from the con-

versation segment. It is especially effective with the modeling of conversation structure, in which case it improves the system's prediction accuracy by an absolute difference of 3.9% on our test data set.

1.2 Outline

The remaining thesis is organized as follows:

- Chapter 2 gives a brief overview of the recent work related to conversation entailment. They are from two areas: 1) textual entailment; and 2) automated processing of conversation scripts.
- Chapter 3 describes a dependency approach to textual entailment.
- Chapter 4 gives a preliminary investigation on conversation entailment.
- Chapter 5 describes our approach to incorporate different conversation features in conversation entailment, including conversation structure.
- Chapter 6 describes the enhanced models for conversation entailment, by incorporating string features to capture semantic relation between language constituents.
- Chapter 7 provides discussions based on our experiments on conversation entailment, unveiling the challenges in the conversation entailment problem.
- Chapter 8 concludes our work and discusses future research directions.

Chapter 2

Related Work

There are two groups of work that are related to conversation entailment: one is in the area of textual entailment, the other concerns various studies based on conversation scripts.

2.1 Textual Entailment

This thesis work is inspired by a large body of recent work on textual entailment initiated by the PASCAL RTE Challenges [8, 10, 22, 36, 37].

Because complete, accurate, open-domain natural language understanding is beyond current capabilities, researchers have attempted to extract the maximum mileage from limited semantic representations. To address the problem of textual entailment, this section gives a brief overview of these approaches.

Perhaps the most common representation of textual content is "bag-of-words" or "bag-of-n-grams" [71]. Based on this representation, simple measures of semantic overlap has been experimented for textual entailment, such as simple overlap counting on bag-of-words or bag-of-n-grams, or weighting by TF-IDF scores, and so on [48]. These models are similar to those typically used in the area of information retrieval

(IR). Treating the text as a document and the hypothesis as a query, the strength of entailment is then assessed by their IR score. However, such models are too impoverished to be of much use, because they do not account for syntactic or semantic information which is essential to determining entailment. For example, the following text-hypothesis pair can get a high IR score, but the hypothesis is not entailed from the text:

> Text: *The National Institute for Psychobiology in Israel was established in 1979.*
> Hypothesis: *Israel was established in 1979.*

Apart from the IR-based approaches, more interesting approaches take into account the structure information in natural language. Based on different representations of the language structure, they can be classified into two major classes: logic-based approaches and graph-based approaches.

2.1.1 Logic-based Approaches

Since the terms *entailment, inference,* and *equivalence* all originated from logic [87], it is perhaps the most natural idea to target this problem by logic proving. By converting the natural language sentences into logic representations, one can decide that the text entails the hypothesis if the hypothesis can be proved from the text.

Logic representations of natural language ranges from traditional first-order logic [1, 32] and Discourse Representation Theory [12] to neo-Davidsonian-style quasi-logical form [65, 76], but they are in essence similar. Take the one used by Raina et al. [76] for example,

> T: *Bob purchased an old convertible.*
> H: *Bob bought an old car.*

can be represented as

$$\text{T: } (\exists A, B, C) Bob(A) \wedge convertible(B) \wedge old(B) \wedge purchased(C, A, B)$$
$$\text{H: } (\exists X, Y, Z) Bob(X) \wedge car(Y) \wedge old(Y) \wedge bought(Z, X, Y)$$

With this representation, the hypothesis is inferred from the text if and only if it can be logically proved from the latter. A strict theorem prover finds a proof for the hypothesis given the text using the method of resolution refutation. It adds the negation of the goal logical formula (i.e., the hypothesis) to a knowledge base consisting of the given axioms (i.e., the text), and then derives a null clause through successive resolution steps. This corresponds to justifying (i.e., "proving") the goal by deriving a contradiction for its negation. For example, the following clauses are obtained for the previous example:

$$(\exists A, B, C) Bob(A) \wedge convertible(B) \wedge old(B) \wedge purchased(C, A, B)$$
$$(\forall X, Y, Z) \neg Bob(X) \vee \neg car(Y) \vee \neg old(Y) \vee \neg bought(Z, X, Y)$$

However, approaches relying on strict logic proving has limited use in practice due to two major reasons. First, they require full understanding of the language and accurate representation of all semantic relations in terms of logic. However, accurate logic representation for natural language is not currently available, and the state-of-the-art semantic parsers extract only some of the semantic relations encoded in a given text. Second, world knowledge is often required in the process of reasoning. For example, one must either know or assume that "*a convertible is a car*" in order to correctly infer the entailment "*Bob bought an old car*" in the previous example. As a result, previous approaches relying on mapping to first order logic representations with a general prover without using rich knowledge sources [12] have not borne much fruit.

Because logic entailment is a quite strict standard, logic-based approaches tend to lead to high precision but low recall [12]. Facing this issue, researchers have been seeking for various compromises to relax the strictness and increase flexibility. The abductive reasoning approach [76] relaxes the unification of logic terms to an approximate one, and encode their knowledge about the semantics into a cost function assessing the plausibility of the approximated unifications. As the function is trained on a labeled set of data statistically, this approach is more robust and scalable and results in higher recall.

To incorporate world knowledge into the logic proving model, some systems employs hand-crafted semantic axioms to enrich the logic representation of natural language before the proving process [65]. This provides an enrichment to the semantic relations, but it is less scalable to be applicable to large data or broader domains.

MacCartney and Manning [58] introduced *natural logic* to model containment and exclusion in the entailment problem. They classified all entailment relations into seven mutually exclusive classes: equivalence (couch = sofa); forward entailment (crow $\sqsubset$ bird) and its converse (European $\sqsupset$ French); negation, or exhaustive exclusion (human $\hat{}$ nonhuman); alternation, or non-exhaustive exclusion (cat | dog); cover, or non-exclusive exhaustion (animal $\smile$ nonhuman); and independence (hungry # hippo). They then form the entailment of a compound expression as a function of the entailments of its parts. Semantic functions $f(\cdot)$ are categorized into different projectivity classes, which describe how the entailment relation between $f(x)$ and $f(y)$ depends on the entailment relation between x and y. For example, simple negation (*not*) projects =, #, and $\hat{}$ without change (*not happy* = *not glad*, *isn't swimming* # *isn't hungry*, and *not human* $\hat{}$ not nonhuman), and swaps $\sqsubset$ and $\sqsupset$ (*didn't kiss* $\sqsupset$ *didn't touch*) and | and $\smile$ (*not French* $\smile$ *not German*, *not more than 4* | *not less than 6*). This allows the system to determine the entailment of a compound expression recursively, by propagating entailments upward through a

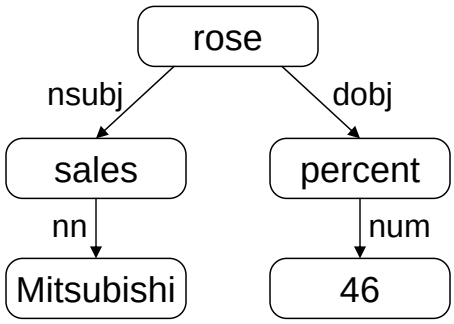


Figure 2.1: An example for dependency graph (from MacCartney et al. [59])

semantic composition tree according to the projectivity class of each node on the path to the root. For example, the semantics of *Nobody can enter without a shirt* might be represented by the tree *(nobody (can ((without (a shirt)) enter)))*. Since *shirt* $\sqsubset$ *clothes*, so *without shirt* $\sqsupset$ *without clothes*, and *Nobody can enter without a shirt* $\sqsubset$ *Nobody can enter without clothes*. As we can see, the judgement of entailment here still follows a rather strict standard. Therefore the system's performance on the PASCAL RTE Challenge resulted in relatively high precision but low recall.

2.1.2 Graph-based Approaches

The graph-based approach is to formulate the entailment prediction as a graph matching problem. It represents the text and the hypothesis as semantic graphs derived from syntactic dependency parses [25, 40]. Figure 2.1 shows an example of the graph representation for a sentence "*Mitsubishi sales rose 46 percent*".

Given the graph representations for both the text and the hypothesis, semantic alignments are performed between the graph representing the hypothesis and a portion of the corresponding graph(s) representing the text. Each possible alignment of the graphs has an associated score, and the score of the best alignment is used as an approximation to the strength of the entailment. Figure 2.2 shows an example of matching the hypothesis "*Bezos established a company*" to the text "*In 1991, Amazon.com was founded by Jeff Bezos*" and the cost of this match.

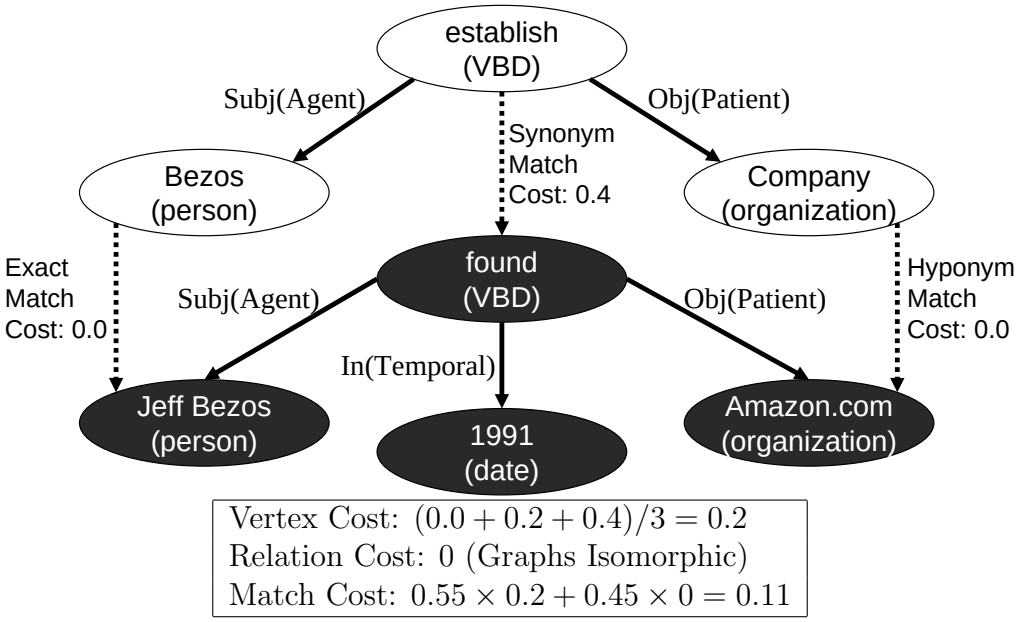


Figure 2.2: An example for graph matching (from Haghighi et al. [40])

MacCartney et al. [59] used a two-stage approach to first find the alignment between the two graphs and then make the entailment prediction. In the first step, the algorithm searches for a good partial alignment from the typed dependency graph representing the hypothesis to the one representing the text, which maximizes the alignment score. In the second step, a classifier was trained to determine the entailment relationship given the complete aligned graph.

MacCartney et al. [60] has taken the alignment step further. Their work aligns phrases in the sentences rather than nodes in the graph (or tokens in the sentences). In their notion, "phrase" refers to any contiguous span of tokens, not necessarily corresponding to a syntactic parse. The phrase-based alignment is to eliminate the needs for many-to-many alignments, since they can be reduced to one-to-one alignments on phrase level. For example, in "*In most Pacific countries there are very few women in parliament.*" and "*Women are poorly represented in parliament.*" they can align *very few* and *poorly represented* as units, without being forced to make a difficult choice as to which word goes with which word.

Because finding the best alignment between two graphs is NP-complete, exact computation is intractable. Therefore researchers have proposed a variety of approximate search techniques, such as local greedy hill-climbing search [40], or incremental beam search [59].

Similar to the semantic axioms [65] in logic-based approaches, de Salvo Braz et al. [25] use "rewriting rules" in the graph-based approach to generate intermediate forms from the original text, with a good supply of additional linguistic and world knowledge axioms. The cost of matching the text to the hypothesis is then determined by the minimal cost among matches from all the intermediate forms to the hypothesis.

Such rewriting rules are also referred to as *inference rules* [27, 55], *entailment rules* [85], or *entailment relations* [86]. They are acquired from large corpora based on the *Distributional Hypothesis* [41]. The *Distributional Hypothesis* states that phrases in similar context tend to have similar meanings. For example, if *X prevents Y* and *X provides protection against Y* are repeatedly seen in a large corpora, it can be induced that *prevent* implies *provide protection against*, and thus *prevent* $\rightarrow$ *provide protection against* is an inference rule. The current largest collection of such rules is DIRT [55]. These rules were widely applied to solve the textual entailment problem [27].

Besides DIRT and other efforts to acquire binary rules (rules templates with two variables) [78, 86], recent work [85] has proposed unsupervised learning of unary rules (e.g., *X take a nap* $\rightarrow$ *X sleep*). However, their applications on the textual entailment task have not yet been explored.

2.1.3 Comparing Logic-based and Graph-based Approaches

Although they use different forms of representations for natural language, logic-based and graph-based approaches are considered isomorphic by MacCartney et al. [59].

In a graph representation, the nodes and edges can be seen as the logic terms in a logic representation. For instance, the graph in Figure 2.1 can be represented in

neo-Davidsonian quasi-logic form as follows:

$$rose(e_1), nsubj(e_1, x_1), sales(x_1), nn(x_1, x_2), Mitsubishi(x_2), dobj(e_1, x_3), \\ percent(x_3), num(x_3, x_4), 46(x_4)$$

In fact, the logic representations are often derived from dependency graphs by a semantic parser.

The alignment between the hypothesis graph and the text graph can be seen as resolving logic terms in logic proving. They both consider matching an individual node or term of the hypothesis with some counter part from the text. And weighting different semantic features in the procedure of calculating the graph matching (or entailment) score is similar to the "abductive reasoning" approach [76], where logic terms are resolved by some score calculated over a set of features.

2.1.4 Performance Analysis

The PASCAL RTE Challenge, which has been held for five times, provides a benchmark for evaluating systems' performance on judging the entailment. Here we give a brief overview of the results of the last three, the third [36], fourth [37], and fifth [10] PASCAL RTE Challenges.

In the RTE-3 task [36], a development set and a test set were provided, each of which contained 800 text-hypothesis pairs. A system's performance was evaluate by its accuracy on the test set, that is, how many entailment relationships (true or false) were correctly predicted out of the 800 pairs. A natural baseline by random guess would obtain 50% accuracy.

There were 45 systems who participated in this evaluation. Among them the best system achieved an accuracy of 80.0%, and the mean and median accuracies were 61.7% and 61.8%, respectively.

It should be well noted from our previous discussion, that the main architectures for different systems are more or less the same. So the critical part that makes the performance difference is how much knowledge is incorporated in the systems. The participating systems in the PASCAL workshop made wide use of various sources of public knowledge bases, such as WordNet [30, 64], DIRT [55], FrameNet [7], VerbNet [51], and PropBank [50]. But the most successful systems [43] (with the highest accuracy) have used additional knowledge sources, including Extended WordNet [47], XWN-KB [88], TARSQI [88], and Cicero/Cicero-Lite [44], most of which were not publicly available. MacCartney et al. [60] indicated that such systems are "idiosyncratic and poorly-documented", "often using proprietary data, making comparisons and further development difficult".

The fourth PASCAL RTE Challenge [37] attracted participation of 45 systems. Their prediction accuracies range from 49.7% to 74.6%, with an average of 57.9% and median of 57.0%. The fifth PASCAL RTE Challenge [10] had participation of 54 systems. The prediction accuracies range from 50.0% to 73.5%, with an average of 61.1% and median of 60.4%. The data collections of these two challenges followed the same setting as the third challenge. Comparing these three evaluations on textual entailment, although different data were actually used to evaluate the participating systems, there are no significant variations in their result statistics.

Among the participating systems in the last three PASCAL RTE Challenges, although some of them have explored very in depth into specific technical aspects (e.g. entailment of temporal expressions [90]), the overall framework of methodology has not evolved much. In other words, they were continuously solving the textual entailment problem either by logic proving or by graph matching.

Nevertheless, a conversation discourse is very different from a written monologue discourse. The conversation discourse is shaped by the goals of its participants and their mutual beliefs. The key distinctive features include turn-taking between par-

ticipants, grounding between participants, and different linguistic phenomena of utterances (e.g., utterances in a conversation tend to be shorter, with disfluency, and sometimes incomplete or ungrammatical). It is the goal of this thesis to explore how techniques developed for textual entailment can be extended to address these unique behaviors in conversation entailment.

2.2 Studies on Conversation Scripts

Recent work has applied different approaches to extract and acquire various kinds of information from human-human conversation scripts. Related work ranges from low-level recognition of conversation structure to high-level applications such as identifying biographical facts, attributes, and social relations, detecting agreements and disagreements between participants, meeting summarization, and predicting success in task-oriented dialogues.

2.2.1 Recognition of Conversation Structures

Related work on recognizing conversation structures based on conversation scripts includes the recognition of dialogue acts and discourse structures.

Dialogue Acts

The ability to model and automatically detect discourse structure is an important step toward dialogue understanding. Dialogue acts are the first level of analysis of discourse structure. A dialogue act represents the meaning of an utterance at the level of illocutionary force [5], such as *Statement*, *Question*, *Backchannel*, *Agreement*, *Disagreement*, and *Apology*. Although specific applications only require relevant dialogue act categories, Allen and Core [3] developed a dialogue act labeling system that is domain-independent.

Stolcke et al. [84] presented a domain independent framework for automated dialogue act identification, which for the most part treats dialogue act labels as a formal tag set. The model is based on treating the discourse structure of a conversation as a hidden Markov model [75]. The HMM states correspond to dialogue acts and the observations correspond to utterances. The features that are used by Stolcke et al. [84] to describe the utterances are mostly based on conversation transcripts, including transcribed words and recognized words from the speech recognizer. But they use some of the prosodic features too, such as pitch, duration, energy, etc.

The HMM representation allows efficient dynamic programming algorithms to compute relevant aspects of the model, such as

- The most probable dialogue act sequence (the Viterbi algorithm).

- The posterior probability of various dialogue acts for a given utterance, after considering all the evidence (the forward-backward algorithm).

The Viterbi algorithm for HMM [89] finds the globally most probable state sequence. When applied to a discourse model, it will therefore find precisely the dialogue act sequence with the highest posterior probability. Such Viterbi decoding is fundamentally the same as the standard probabilistic approaches to speech recognition [6] and tagging [19].

While the Viterbi algorithm maximizes the probability of getting the *entire* dialogue act sequence correct, it does not necessarily find the dialogue act sequence that has the most dialogue act labels correct [26]. To maximize the total accuracy of utterance labeling, it is needed to maximize the probability of getting each dialogue label correct individually, which can be efficiently carried out by the forward-backward algorithm for HMM [9].

Opinion Frames

Opinions in conversations are defined [80, 91] in two classes: *sentiment* includes positive and negative evaluations, emotions, and judgments; and *arguing* includes arguing for or against something, and arguing that something should or should not be done. Opinions have a *polarity* that can be positive or negative. The *target* of an opinion is the entity or proposition that the opinion is about. For example (a conversation about designing a remote control, from Somasundaran et al. [82]):

C: ... shapes should be curved, so round shapes. Nothing square-like.

⋮

C: ... So we shouldn't have too square corners and that kind of thing.

B: Yeah okay. Not the old box look.

In the utterance "*shapes should be curved*" there is a positive argument with the target *curved*, and in the utterance "*Not the old box look*" there is an negative sentiment, with the target *the old box look*.

It is argued that while recognizing opinions of individual expressions and their properties is important, discourse interpretation is needed as well [82]. In the above example, we see from the discourse that *curved*, *round shapes* are the preferred types of design, and *square-like*, *square corners*, and *the old box look* are not.

The discourse level association of opinions are modeled as *opinion frames* [82]. An opinion frame consists of two opinions that are related by virtue of having related targets. There are two relations between targets, *same* and *alternative*. The *same* relation holds between targets that refer to the same entity, property, or proposition. Here the term "same" covers not only identity, but also part-whole, synonymy, generalization, specialization, entity-attribute, instantiation, cause-effect, epithets and implicit background topic. The *alternative* relation holds between targets that are

related by virtue of being opposing (mutually exclusive) options in the context of the discourse. In the above example, there is an *alternative* relation between targets *curved* and *square-like*, and there are *same* relations between targets *square-like*, *square corners*, and *the old box look*.

An *opinion frame* is defined as a structure composed of two opinions and their respective targets connected via their target relations. For each of the two opinion slots, there are four possible type/polarity combinations (sentiment/arguing combined with positive/negative). So combined with two possible target relations (same/alternative), there are totally $4 \times 4 \times 2 = 32$ different types of opinion frames. In the above example, *shapes should be curved* and *Nothing square-like* constitutes an opinion frame of APANalt (positive arguing and negative arguing with alternative targets).

Somasundaran et al. [82] argued that recognizing opinion frames will provide more opinion information for NLP applications than recognizing individual opinions alone, because opinions regarding something not lexically or even anaphorically related can become relevant. Take the *alternative* relation for instance, opinions towards one alternative can imply opinions of opposite polarity toward the competing options. In the above conversation example, if we consider only the explicitly stated opinions, there is only one (positive) opinion about the *curved shape*. However, the speaker expresses several other (negative) opinions about alternative shapes, which reinforce his positivity toward the curved shape. Thus, by using the frame information, it is possible to gather more opinions regarding curved shapes for TV remotes.

Further, if there is uncertainty about any one of the components, they believe opinion frames are an effective representation incorporating discourse information to make an overall coherent interpretation [46]. In particular, suppose that some aspect of an individual opinion, such as polarity, is unclear. If the discourse suggests certain opinion frames, this may in turn resolve the underlying ambiguity. Again in the above example, the polarity of *round shapes* may be unclear. However, the polarity

of *curved* is clear, and by recognizing there is a *same* relation between these two targets, it is possible to resolve the ambiguity in the polarity of *round shapes*, which is also positive.

Somasundaran et al. [82] proposed a machine learning approach to detect opinion frames. This is formulated as a classification problem: given two opinion sentences, determine if they participate in any frame relation. Their experiments assume oracle opinion and polarity information, and consider frame detection only between sentence pairs belonging to the same speaker. The data used in their work is the AMI meeting corpus [16], with annotations [81] for sentiment and arguing opinions (text anchor and type). A variety of features including content word overlap, focus space overlap, anaphoric indicator, time difference, adjacency pair, and standard bag of words were used in their experiment to determine if two opinions are related.

Somasundaran et al. [83] used the opinion frames to improve the polarity classification of opinions. In their work they first implemented a local classifier to bootstrap the classification process, and then implemented classifiers that use discourse information (i.e., opinion frames) over the local classifier. They explored two approaches for implementing the discourse-based classifier:

1. Iterative Collective Classification [56, 69]: instances are classified in two phases, the bootstrapping phase and the iterative phase. In the bootstrapping phase, the polarity of each instance is initialized to the most likely value given only the local classifier and its features. In the iterative phase, discourse relations and the neighborhood information brought in by these relations are incorporated as features into a relational classifier.

2. Integer Linear Programming: the prediction of opinion polarity is formulated as an optimization problem, which maximizes the class distributions predicted by the local classifier, subject to constraints imposed by discourse relations.

2.2.2 High Level Applications

Recent work has studied multiple types of specific inference that can be made from conversation scripts. These include biographic attributes, social networks and biographical facts, agreements and disagreements, summarization, and success in task-oriented dialogues.

Latent Biographic Attributes

Biographic attributes of conversation speakers include gender, age, and native/non-native speaker. Such information is derivable from acoustic properties of the speaker, including pitch and f0 contours [11]. Recently, however, Garera and Yarowsky [35] worked on modeling and classifying such speaker attributes from only the latent information found in conversation scripts. In particular, they modeled and classified biographic attributes such as gender and age based on lexical and discourse factors including lexical choice, mean utterance length, patterns of participation in the conversation and filler word usage.

Garera and Yarowsky [35] built their work upon the previous state-of-the-art [13], which models gender of speakers using unigram and bigram features in an SVM framework. For each conversation participant, they created a training example using unigram and bigram features with tf-idf weighting, as done in standard text classification approaches. Then an SVM model was trained to learn the weights associated with the n-gram features. They found some of the gender-correlated words proposed by sociolinguistics are also assigned with more discriminative weights by this empirical model, such as the frequent use of "oh" by females. They evaluated the performance of their approach on the Fisher telephone conversation corpus [23] and the standard Switchboard conversational corpus [38].

Garera and Yarowsky [35] further argued that a speaker's lexical choice and discourse style may differ substantially depending on the gender, age, and dialect of the

other person in the conversation. The hypothesis is that people tend to use stronger gender-specific, age-specific or dialect-specific word, phrase and discourse properties when speaking with someone of a similar gender, age, or dialect, compared to speaking with someone of a different gender, age, or dialect. In the latter case, they may adapt a more neutral speaking style. So Garera and Yarowsky [35] proposed to add performance gains in gender classification by using a stacked model conditioning on the predicted partner class. They trained several classifiers identifying the gender of each speaker, the gender combination of the entire conversation, and the conditional gender prediction of each speaker given the most likely gender of the other speaker. They then used the score of each classifier as a feature in a meta SVM classifier.

There has also been substantial work in the sociolinguistics literature investigating discourse style differences due to speaker properties such as gender [20, 29]. Those works have shown gender differences for speakers due to features such as speaking rate, pronoun usage and filler word usage, suggesting that non-lexical features can further help improve the performance of gender classification on top of the standard n-gram model. Garera and Yarowsky [35] investigated a set of features such as speaker rate and percentage of pronoun usage, motivated by the sociolinguistic literature on gender differences in discourse [57].

Garera and Yarowsky [35] also extended their approach on gender classification to the prediction of speakers' age and native/non-native speaker. Again they had findings consistent with the sociolinguistic studies for age [57], such as frequent usage of the word "well" among older speakers.

Social Networks and Biographical Facts

Jing et al. [49] gave a framework to extract social networks and biographical facts from conversation speech transcripts. Entities, relations, and events are extracted separately from the conversation scripts by different information extraction modules,

and a fusion module is then used to merge their outputs and extract social networks and biographical facts.

Identified person entities and extracted relations are fused as nodes and ties in a social network. For example, from the input sentence *my mother is a cook*, a relation detection system identifies the relation $motherOf(mother, my)$. And if an entity recognition module identifies that *my* refers to the person *Josh* and *mother* refers to the person *Rosa*, then by replacing *my* and *mother* with the corresponding named entities, the fusion module produces the following nodes and ties in a social network: $motherOf(Rosa, Josh)$.

As can be seen from this example, coreference resolution plays a critical role in the extraction of social networks. As a result, Jing et al. [49] paid a major effort on improving coreference resolution for conversations, by both feature engineering and improving the clustering algorithm.

Biographical facts are extracted in a similar way by selecting the events (extracted by the event extraction module) and corresponding relations (extracted by the relation extraction module) that involve a given individual as an argument.

Agreements and Disagreements

Conversations involve many agreements and disagreements of one speaker to another. Galley et al. [34] focused on the identification of agreements and disagreements on the utterance level, and formulated the problem as a multi-class classification problem: given an utterance from a speaker, the task is to classify whether it is an agreement, a disagreement, or none of these two. They suggested to use a sequence classification model to approach this task, with a set of local and contextual features characterizing the occurrence of agreements and disagreements.

The local features include lexical features such as agreement markers [21], e.g. *yes* and *right*, general cue phrases [45], e.g. *but* and *alright*, and adjectives with positive

or negative polarity [42]. A set of durational features are also incorporated and described as good predictors of agreements: utterance length distinguishes agreement from disagreement. The latter tends to be longer since the speaker elaborates more on the reasons and circumstances of her disagreement than for an agreement [21]. And a fair amount of silence and filled pauses is sometimes an indicator of disagreement, since it is a dispreferred response in most social contexts and can be associated with hesitation [73].

Galley et al. [34] also noted that context provides important information to the classification of agreements and disagreements. For example, whether an utterance is an agreement or a disagreement is largely influenced by whether the previous utterance from the same speaker is an agreement or a disagreement, i.e. an agreement is more likely to be followed by another agreement, and vice versa. There are also reflexive and transitive contexts that may be indicative. Reflexivity means if A disagrees with B, then B is also likely to disagree with A. Transitivity means, for example, if A agrees with B and B disagrees with C, then A may also disagrees with C, and so forth.

In order to capture both the local and the contextual features to classify the agreements and disagreements, Galley et al. [34] used a Bayesian network to perform the classification. The most probable agreement/disagreement sequence is computed by performing a sequential decoding with beam search.

Meeting Summarization

Automatic summarization helps the processing of information contained in conversation scripts. Murray and Carenini [66] took an extractive approach to conversation summarization. They conducted a binary classification on sentences in a conversation, identifying whether each sentence should be extracted as the summary. Sentences were ranked by their classification scores, and a top portion of sentences were

kept as the conversation summary until they reach a certain threshold of word count.

To locate the most salient sentences in a conversation, Murray and Carenini [66] derived various features to train their classifier, which include sentence lengths that were previously found to be effective in speech and text summarization [33, 62, 67], structural features capturing the relation between a sentence and the conversation, features related to conversation participants, a lexical feature capturing varying interests and expertise between the conversation participants, a lexical feature capturing topic shifts in a conversation, cosine features capturing whether the conversation is changed by a sentence in some fashion, centroid features capturing the similarity between a sentence and the conversation, word entropy features measuring how informative a sentence is, and whether a sentence is a turning point in the conversation, and the ClueWordScore used by Carenini et al. [15].

Murray and Carenini [66] used a simple feature subset selection based on the F statistics [18], and applied their extractive summarization system to a portion of the AMI corpus [16]. They found that the best features for summarization are sentence length, sum of term scores (described above), and the centroid features that measure whether the candidate sentence is similar to the conversation. Their evaluation results show that such a summarization system, which relies solely on features extracted from conversation scripts, achieved a competitive performance compared to the state-of-the-art summarization systems that also employ speech-specific (e.g. prosodic) features. Therefore, the same summarization system is also applicable to other domains similar to spoken conversations, such as email threads.

Predicting Success in Task-oriented Dialogues

In task-oriented dialogues, an important indicator of the communication effectiveness is whether the task is accomplished successfully.

Pickering and Garrod [72] suggested in their Interactive Alignment Model that dialogues between humans are greatly aided by aligning representations on several linguistic and conceptual levels. This effect is assumed to be driven by a cascade of linguistic repetition effects, where interlocutors tend to re-use lexical, syntactic and other linguistic structures after their introduction. Reitter and Moore [77] referred to this repetition effect, or a tendency to repeat linguistic decisions, as *priming*. Motivated the hypothesis of Pickering and Garrod [72], Reitter and Moore [77] deduced that "the connection between linguistic persistence or priming effects and the success of dialogue is crucial" for the Interactive Alignment Model. Based on this assumption, Reitter and Moore [77] proposed an automatic method of measuring task success.

Reitter and Moore [77] tried to predict task success from a dialogue using lexical and syntactic repetition information. They used the HCRC Map Task corpus [4], where subjects were given two slightly different maps and one of them gives directions of a pre-defined route to another. The task success is then determined by the deviation between the route given by the leader and the route followed by the follower, which is measured by the area covered in between the two paths (PathDev). They trained an SVM regression model, using features of lexical, syntactic, and string repetitions and the PathDev score as output. Their results show that "linguistic repetition serves as a good predictor of how well interlocutors will complete their joint task" [77].

Reitter and Moore [77] further compared the indications of short-term priming and long-term priming (alternatively called *adaptation*). It was argued that short- and long-term adaptation effects may be due to separate cognitive processes [31], so they wanted to find out whether alignment in dialogues is due to the automatic, classical priming effect, or whether it is based on a long-term effect that is possibly closer to implicit learning [17].

Through similar experiments using PathDev as a measurement of task success, Reitter and Moore [77] found that path deviation and short-term priming did not

correlate. Despite the fact that priming effect is clear in the short term, "the size of this priming effect does not correlate with task success" [77]. In contrast, there is a reliable correlation of task success and long-term adaptation. Stronger path deviations relate to weaker adaptation. The more adaptation were observed, the better performance were achieved by the subjects in synchronizing their routes on the maps. This confirms their assumption derived from Interactive Alignment Model.

In conclusion, the correlation shows that, of the repetition effects included in the task-success prediction model, it is long-term adaptation as opposed to the more automatic short-term priming effect that contributes to prediction accuracy. "Long-term adaptation may thus be a strategy that aids dialogue partners in aligning their language and their situation models." [77]

Chapter 3

A Dependency Approach to Textual Entailment

Conversations are not completely different from text. After all, a conversation is made up by similar linguistic components, from words, sentences, to discourse. The first question is, to what degree that methods for textual entailment can be used to infer knowledge from conversations.

In this chapter, we describe a dependency-based approach for textual entailment, which provides a reasonable baseline for our investigation on conversation entailment.

3.1 A Framework of the Dependency Approach

As introduced in Chapter 1, a definition of the textual entailment problem is given by the PASCAL RTE Challenge [8, 10, 22, 36, 37]: given a piece of text T and a hypothesis H, the goal is to determine whether the meaning of H can be sufficiently inferred from T.

Formally, we use the sign $\models$ to denote the entailment relationship. We represent

that T entails H as

$$T \vDash H$$

Similarly, if T does not entail H, we represent it as

$$T \nvDash H$$

Given such context, we will use the phrase **premise discourse** to refer to the text from which the meaning is to be inferred (in conversation entailment it is a conversation segment), and use the letter D to denote it. And for the hypothesis, it is usually a single statement (e.g., in the PASCAL RTE data set and our data set). We call it the **hypothesis statement**, and use the letter S to denote it. Thus a generic form of the textual or conversation entailment problem is stated below:

> Given a premise discourse D and a hypothesis statement S, estimate the probability
>
> $$P(D \vDash S|D, S)$$

The probability represents the likelihood of the entailment relationship between D and S, and we can say that D entails S if this likelihood is above a certain threshold (usually 0.5).

3.1.1 Representation

This section discusses how we represent natural language text and statements in our system.

We first introduce several concepts that we are using throughout the presentation of our framework:

- A *term* refers to either an entity or an event:

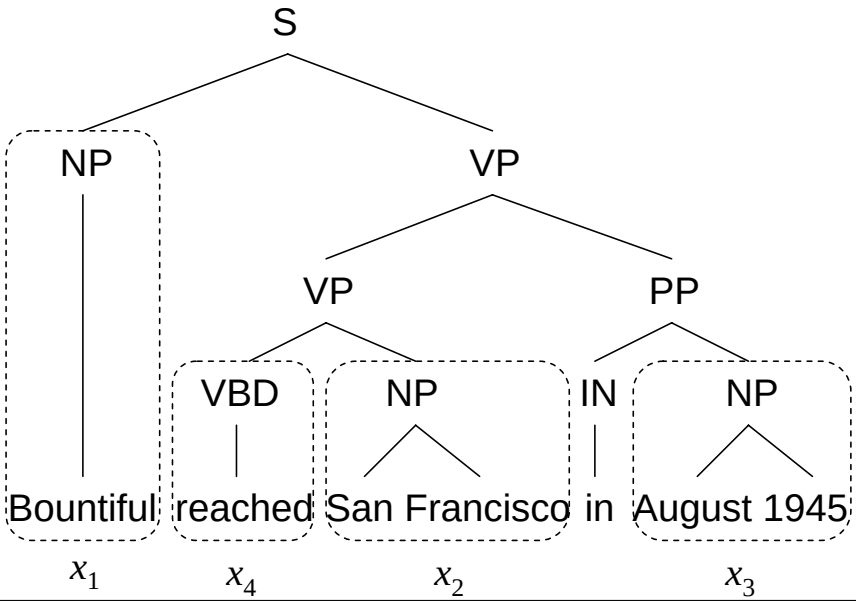
- An *entity* refers to a person, a place, an organization, or other real world entities. This follows the same idea as the concept of *mention* in the Automatic Content Extraction (ACE) Workshops [28]: a *mention* is a reference to a real world entity; it can be named (e.g. *John Lennon*), nominal (e.g. *mother*), or pronominal (e.g. *she*).
- An *event* refers to an action, an activity, or other real world events. For example, from the sentence *John married Eva in 1940* we can identify the event of marriage.

We use lower-case letters to represent terms (e.g., $x = John$, $y = marry$, etc.).

- A *clause* is either a property or a relation:
  - A *property* is a property associated with a term (entity or event). For example, an entity *company* can have a property of *Russian*, and an event *visit* can have a property of *recently*. We use a unary predicate $p(x)$ to represent a property, e.g. $Russian(company)$, $recently(visit)$.
  - A *relation* is a relation between two terms (either entities or events). For example, from the phrase *headquarter in Canada* we can recognize that the entities *headquarter* and *Canada* have a relation of "*is in*". From the phrase *Prime Minister visited Brazil* we can recognize that the event *visit* and the entity *Prime Minister* have a relation that *Prime Minister* "*is the subject of*" *visit*. We use a binary predicate $r(x, y)$ to represent a relation, e.g. $in(headquarter, Canada)$, $subj(visit, Prime\ Minister)$.

Syntactic Decomposition

The clause representation of a natural language sentence is derived from its syntactic parse tree. The process of converting a parse tree to the clause representation can be seen as a decomposition of the tree structure.



Syntactic substructure	Head	Derived clause
NP → Bountiful	x_1 =Bountiful	
NP → San Francisco	x_2 =San Francisco	
NP → August 1945	x_3 =August 1945	
VBD → reached	x_4 =reached	
PP → IN NP	*in* x_3	
VP → VBD NP	x_4	$object(x_4, x_2)$
VP → VP PP	x_4	$preposition(x_4, in\ x_3)$
S → NP VP	x_4	$subject(x_4, x_1)$

Figure 3.1: An example of syntactic decomposition

Decomposing a syntactic parse tree into a set of clauses is based on dependency parsing [24], where a set of hand-crafted rules, or patterns, are applied on the phrase structures. Appendix A lists the set of rules that we developed to derive the dependency structures.

Figure 3.1 illustrates the decomposition process for the statement *Bountiful reached San Francisco in August 1945*. For each phrase structure in the parse tree (e.g., $S \rightarrow NP\ VP$), an associated decomposition rule is used to specify two types of information: (1) the head term of the parent node (e.g., S), which is obtained from one of its children (in this case the head of S is get from the head of VP); (2) the clauses that are to be generated, e.g., for $S \rightarrow NP\ VP$ we generate $subject(h_2, h_1)$, where

h_1 is the head term of the first child (NP), and h_2 is the head term of the second child (VP). The head terms of NP and VP are obtained recursively by decomposition rules defined upon the substructures spanning them (e.g., $NP \rightarrow$ *Bountiful* and $VP \rightarrow$ VP PP, respectively).

We have also taken care of the following processes in our decomposing rules similar to those in dependency parsing [24]:

- Collapsing a prepositional relation $preposition(x, prep\ y)$ into a relational clause between x and y described by *prep*. For example, in Figure 3.1, the clause $preposition(x_4, in\ x_3)$ are collapsed into $in(x_4, x_3)$.

- Processing conjunct dependencies to produce a representation closer to the semantics. For example, for "*bills on ports and immigration*" we produce $on(bills, ports)$ and $on(bills, immigration)$ (as opposed to $on(bills, ports)$ and $and(ports, immigration)$). This is implemented by the multi-head mechanism encoded in our decomposition rules.

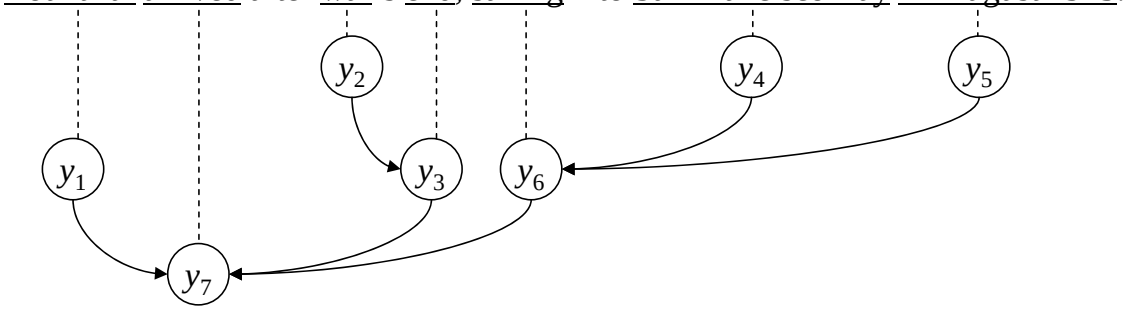
- Adding arguments for relative clauses. e.g. For *I like the man who tells jokes* we have $subject(tells, man)$.

After the syntactic decomposition, both the premise discourse and the hypothesis statement are represented as sets of terms (e.g., $x_1 = Bountiful$, $x_4 = reached$, etc.) and clauses (e.g., $object(x_4, x_2)$, $in(x_4, x_3)$, etc.). Figure 3.2(a) shows an example of a premise and the corresponding hypothesis. Figure 3.2(c) shows the decomposed terms and clauses for the premise and Figure 3.2(e) shows the decomposed representation for the hypothesis.

We use the term "clause" here because logically, a statement is the conjunction of a set of clauses. Similarly a natural language statement can be viewed as a conjunction of clauses defined above.

Premise:	Bountiful arrived after war's end, sailing into San Francisco Bay 21 August 1945.
Hypothesis:	Bountiful reached San Francisco in August 1945.

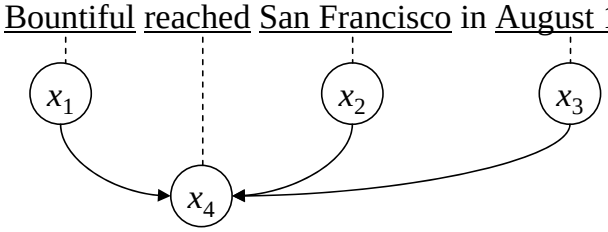
(a) The text premise and hypothesis statement



(b) Dependency structure for the premise

Terms	Clauses
$y_1 = Bountiful$, $y_2 = war$, $y_3 = end$, $y_4 = San\ Francisco\ Bay$, $y_5 = 21\ August\ 1945$, $y_6 = sailing$, $y_7 = arrived$	$modifier(y_3, y_2)$, $into(y_6, y_4)$, $adverbial(y_6, y_5)$, $subject(y_7, y_1)$, $after(y_7, y_3)$, $adverbial(y_7, y_6)$

(c) Clause representation for the premise



(d) Dependency structure for the hypothesis

Terms	Clauses
$x_1 = Bountiful$, $x_2 = San\ Francisco$, $x_3 = August\ 1945$, $x_4 = reached$	$subject(x_4, x_1)$, $object(x_4, x_2)$, $in(x_4, x_3)$

(e) Clause representation for the hypothesis

Figure 3.2: The decomposition of a premise-hypothesis pair

This representation is similar to the neo-Davidsonian-style quasi-logical form [65, 76]. And we also follow its idea of reifying the verb terms. Alternatively, a representation without reification would put the sentence "*Bountiful reached San Francisco*" as $reach(Bountiful, San\ Francisco)$, but in this way the modifier "in August 1945" will have no place unless higher-order logic is introduced.

This representation is also similar to a typed dependency structure, if we view terms as nodes, property clauses as node properties, and relation clauses as dependency edges. The only difference between our representation and a dependency structure is that we only take nouns and verbs as terms (or nodes), and put other words like adjectives and adverbs as properties, e.g. instead of $mod(visit, recently)$ we have $recently(visit)$. Figure 3.2(b) and 3.2(d) show the dependency structures of both the premise and the hypothesis (corresponding to the clause representations in Figure 3.2(c) and 3.2(e), respectively).

3.1.2 The Alignment Model

As both the premise and the hypothesis are represented as terms and clauses:

$$D = \{y_1, \ldots, y_b, d_1(\ldots), \ldots, d_m(\ldots)\}$$

$$S = \{x_1, \ldots, x_a, s_1(\ldots), \ldots, s_n(\ldots)\}$$

where $x_1, \ldots, x_a$ are the terms in the hypothesis, $y_1, \ldots, y_b$ are the terms in the premise, $s_1, \ldots, s_n$ are the clauses in the hypothesis, and $d_1, \ldots, d_m$ are the clauses in the premise, in order to predict whether the hypothesis can be inferred from the premise, we need first to find an association between the terms in the premise and ther terms in the hypothesis. For example, in Figure 3.2, we need to know that the term x_1 in the hypothesis (*Bountiful*) refers to the same entity as term y_1 in the

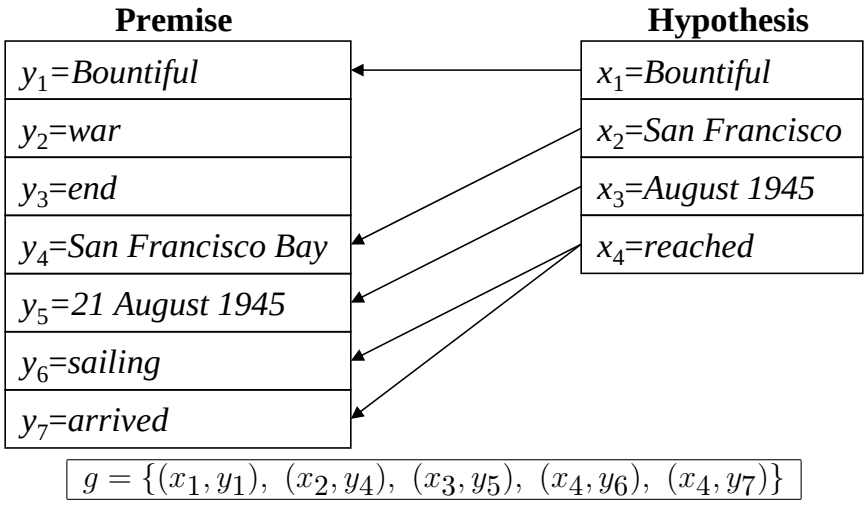


Figure 3.3: An alignment for the example in Figure 3.2

premise (*Bountiful*), and the term x_4 in the hypothesis refers to an event (*reached*) that may be the same as what y_7 refers to in the premise (*arrived*).

Formally, we define an **alignment** g to be a binary relation, i.e., a subset of the Cartesian product, between the hypothesis term set $\{x_1, \ldots, x_a\}$ and the premise term set $\{y_1, \ldots, y_b\}$. A term pair (x, y) is considered to be aligned, i.e., $(x, y) \in g$, if and only if they refer to the same entity or event. Figure 3.3 shows such an alignment for the example in Figure 3.2.

Alternatively, an alignment g can be considered as a binary function defined over a hypothesis term x and a premise term y:

$$g : \{x_1, \ldots, x_a\} \times \{y_1, \ldots, y_b\} \rightarrow \{0, 1\}$$

$$g(x, y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are aligned} \\ 0 & \text{otherwise} \end{cases}$$

Straightforwardly, the function notation of alignment is equivalent to the relation notation:

$$g(x, y) = 1 \equiv (x, y) \in g$$

In this thesis we will use these two notations interchangeably.

Note that an alignment can be between an entity (noun) and an event (verb), e.g. $g(sale, sell) = 1$, or vice versa. It is also possible that one hypothesis term is aligned to multiple premise terms, e.g., (x_4, y_6) and (x_4, y_7) in Figure 3.3, or vice versa.

An **alignment model** θ_A gives such an alignment for any premise-hypothesis pair:

$$\theta_A : D, S \rightarrow g$$

3.1.3 The Inference Model

We have formulated the problem of predicting whether a hypothesis S can be inferred from a premise D as estimating the probability

$$P(D \vDash S|D, S)$$

Suppose we have decomposed the premise D into m clauses $d_1, d_2, \ldots, d_m$ and the hypothesis S into n clauses $s_1, s_2, \ldots, s_n$, the probability to be estimated becomes

$$\begin{aligned} P(D \vDash S|D, S) &= P(D \vDash S|D = d_1 d_2 \ldots d_m, S = s_1 s_2 \ldots s_n) \\ &= P(d_1 d_2 \ldots d_m \vDash s_1 s_2 \ldots s_n | d_1, d_2, \ldots, d_m, s_1, s_2, \ldots, s_n) \end{aligned}$$

Since a statement is the conjunction of the decomposed clauses, whether it can be inferred from a premise is equivalent to whether all of it clauses are inferred from the premise:

$$P(D \vDash s_1 s_2 \ldots s_n | D, s_1, s_2, \ldots, s_n) = P(D \vDash s_1, D \vDash s_2, \ldots, D \vDash s_n | D, s_1, s_2, \ldots, s_n)$$

And to simplify the problem, we make the assumption that whether a clause is

entailed from the premise is conditionally independent from other clauses. So

$$P(D \vDash s_1, D \vDash s_2, \ldots, D \vDash s_n | D, s_1, s_2, \ldots, s_n) = \prod_{j=1}^{n} P(D \vDash s_j | D, s_j)$$

And the probability to be estimated is given by the following formula

$$\begin{aligned} P(D \vDash S | D, S) &= \prod_{j=1}^{n} P(D \vDash s_j | D = d_1 d_2 \ldots d_m, s_j) \\ &= \prod_{j=1}^{n} P(d_1 d_2 \ldots d_m \vDash s_j | d_1, d_2, \ldots, d_m, s_j) \end{aligned} \tag{3.1}$$

An **inference model** θ_E gives such probabilities, that whether a clause from the hypothesis is entailed by a set of clauses from the premise, given an alignment g between the terms in the hypothesis and the terms in the premise, i.e.

$$\theta_E : d_1, d_2, \ldots, d_m, s_j, g \rightarrow P(d_1 d_2 \ldots d_m \vDash s_j)$$

And from Equation (3.1) we know that given a premise-hypothesis pair and an instance of alignment, the inference model also gives the probability that the hypothesis is inferred from the premise:

$$\theta_E : D, S, g \rightarrow P(D \vDash S)$$

3.2 Learning the Entailment Models

With the dependency-based framework consisting of two-stage models, the alignment model and the inference model, next we describe how we build these models.

In the PASCAL RTE data sets [8, 10, 22, 36, 37], for every entailment example we have the truth judgement of whether the hypothesis can be inferred from the

premise given by human annotators. Furthermore, work has been done on manually annotating the word-level alignments for the RTE-2 data set [14]. Therefore, it is natural to adopt the machine learning methodology and learn our entailment models from those annotated data. Particularly, we train both the alignment and inference models using a machine-learning framework.

3.2.1 Learning the Alignment Model

Recall from Section 3.1.2 that an alignment model gives the alignment for a premise-hypothesis pair (D, S):

$$\theta_A : D, S \rightarrow g$$

That is, for each term in the hypothesis x and each term in the premise y, it gives

$$\theta_A : x, y \rightarrow g(x, y)$$

This is a binary classification problem: given a term pair (x, y), we want to make the binary decision of the value of $g(x, y)$ (0 or 1).

We propose to use a feature vector $\mathbf{f}_A(x, y)$ to characterize the lexical, structural, and semantic features of the terms x and y, and use a binary classification model to estimate their alignment score, $g(x, y)$. We can use the notation θ_A to refer to this classification model:

$$\theta_A : \mathbf{f}_A(x, y) \rightarrow g(x, y)$$

To train such a classification model, we consider a training set with a gold-standard alignment g^* for each entailment pair (D, S). Given such a training set $\mathbf{T}$, we can learn an alignment model by maximizing the log-likelihood of the aligned term pairs

(positive training instances):

$$\sum_{(D,S,g^*)\in\mathbf{T}} \sum_{(x,y)\in g^*} \log P(g(x,y)=1|D,S,\theta_A)$$

and minimizing the log-likelihood of unaligned term pairs (negative training instances):

$$\sum_{(D,S,g^*)\in\mathbf{T}} \sum_{(x,y)\notin g^*} \log P(g(x,y)=1|D,S,\theta_A)$$

Thus the learned model θ_A maximizes the log-likelihood of predicting the gold-standard alignments:

$$\sum_{(D,S,g^*)\in\mathbf{T}} \log P(g=g^*|D,S,\theta_A)$$

3.2.2 Learning the Inference Model

Recall from Section 3.1.3 that an inference model gives the probability that a clause from the hypothesis, s_j, is entailed by a set of clauses from the premise, $d_1, d_2, \ldots, d_m$, given an alignment g between the terms in the hypothesis and the terms in the premise:

$$\theta_E : d_1, d_2, \ldots, d_m, s_j, g \rightarrow P(d_1 d_2 \ldots d_m \vDash s_j)$$

As in the alignment case, here we also formulate the inference prediction as a binary classification problem: we first use a feature vector $\mathbf{f}_E(d_1, d_2, \ldots, d_m, s_j, g)$ to characterize the lexical, structural, and semantic features of the clauses $d_1, d_2, \ldots, d_m, s_j$ given the alignment g, and then build a classification model θ_E to estimate the probability $P(d_1 d_2 \ldots d_m \vDash s_j)$ given such a feature vector:

$$\theta_E : \mathbf{f}_E(d_1, d_2, \ldots, d_m, s_j, g) \rightarrow P(d_1 d_2 \ldots d_m \vDash s_j)$$

Again we use the same notation θ_E for the classification model here because of its equivalence to the original inference model.

Now we want to train such a model θ_E from a data set of positive entailment examples, $\mathbf{T}^+ = \{(D, S)^+\}$, where the premises entail the corresponding hypotheses, and a data set of negative entailment examples, $\mathbf{T}^- = \{(D, S)^-\}$, where the premises do not entail the corresponding hypotheses. We follow the assumption that for each entailment example (D, S), we have a gold-standard alignment g^*. Additionally, we also assume that for each of the hypothesis clauses s_j, we have the ground truth that whether it is entailed from the premise D, given the gold-standard alignment g^*.

We use $S^+(D, S, g^*)$ to denote the set of clauses in S that are entailed from D given g^* (positive training instances), and $S^-(D, S, g^*)$ to denote the set of clauses in S that are not entailed from D given g^* (negative training instances). Then an inference model can be learned to maximize the log-likelihood:

$$\sum_{(D,S,g^*)\in\mathbf{T}^+} \sum_{s_j\in S} \log P(D \vDash s_j | D, s_j, g^*, \theta_E) +$$
$$\sum_{(D,S,g^*)\in\mathbf{T}^-} \sum_{s_j\in S^+(D,S,g^*)} \log P(D \vDash s_j | D, s_j, g^*, \theta_E) +$$
$$\sum_{(D,S,g^*)\in\mathbf{T}^-} \sum_{s_j\in S^-(D,S,g^*)} \log P(D \nvDash s_j | D, s_j, g^*, \theta_E)$$

Note that for $\mathbf{T}^+$, $S^+(D, S, g^*) = S$ and $S^-(D, S, g^*) = \phi$ (every clause in the hypothesis should be entailed from the premise).

As such, a learned model θ_E also maximizes the log-likelihood of giving the right entailment judgement for each premise-hypothesis pair:

$$\sum_{(D,S,g^*)\in\mathbf{T}^+} \log P(D \vDash S | D, S, g^*, \theta_E) + \sum_{(D,S,g^*)\in\mathbf{T}^-} \log P(D \nvDash S | D, S, g^*, \theta_E)$$

3.3 Feature Design

Section 3.2 gave the framework of learning the alignment and inference models from annotated data set. This section discusses the indicative features that are used in learning these models.

3.3.1 Features for the Alignment Model

As introduced in Section 3.2.1, a feature vector for the alignment model $\mathbf{f}_A(x, y)$ is defined over a term x from the hypothesis and a term y from the premise.

In theory, a verb term and a noun term can potentially be aligned together. However, to simplify the problem, here we restrict the problem to the alignment between two nouns or two verbs. We designed different feature sets according to whether x and y are nouns or verbs.

Features for Noun Term Alignment

If x and y are noun terms, the feature vector $\mathbf{f}_A(x, y)$ is composed by:

1. String equality: whether the string forms of x and y are equal.

2. Stemmed equality: whether the stems of x and y are equal.

3. Acronym equality: whether one term is the acronym of the other, e.g., *Michigan State University* and *MSU*.

4. Named entity equality: whether two names refer to the same entity, e.g., *President Obama* and *Barack Obama* are the same person. Our simple approach to estimate the equivalence of two named entities is by comparing the right-most terms in the two names (e.g., *Obama* in the above example).

5. WordNet similarity [54]: a similarity measurement of the two terms based on the WordNet taxonomy:

$$sim_W(x, y) = \frac{2 \times \log P(C_{xy})}{\log P(C_x) + \log P(C_y)}$$

where C_x is the WordNet class containing x, C_y is the WordNet class containing y, C_{xy} is the most specific class that subsumes both C_x and C_y, and $P(C)$ is the probability that a randomly selected object belongs to C.

6. Distributional similarity: a similarity measurement of the two terms based on the Dice coefficient of their distributions in a large text corpus:

$$sim_D(x, y) = \frac{2\left|D_x \cap D_y\right|}{|D_x| + \left|D_y\right|}$$

where D_x is the set of documents that contain the term x, and D_y is the set of documents that contain the term y. We use the AQUAINT [39] news corpus as the document collection here.

Features for Verb Term Alignment

To learn the alignment model for verb terms, we use most of the features that are similar to those in the noun alignment model, including string equality, stemmed equality, WordNet similarity, and distributional similarity. However, we also designed a few more features specialized to verb alignment. One of these features is the verb *be* identification, which identifies whether any of the two verbs, x from the hypothesis and y from the premise, is any form of the verb *be*:

$$f_{vb}(x, y) = \begin{cases} 1 & \text{if both or neither of } x \text{ and } y \text{ is verb } be \\ 0 & \text{otherwise} \end{cases}$$

Further more, for an action/event, it is identified by not only the class or type of the action/event, which is described by the verb, but also the executer and receiver of the action or participators in the event, which are described by the verb's arguments. Here we consider two types of arguments: subject and object.

- Two action/events are not the same if their **subjects** (when present) are different, e.g., *A laughed* and *B laughed*;
- Two action/events are not the same if their **objects** (when present) are different, e.g., *A watched TV* and *A watched a football game*.

Note that action/events could be identified by other arguments or adjuncts too. For example, temporal phrases as in *A went to New York in 1970* and *A went to New York last week*. Here, we take a consistent approach that only identifies the action/events by the verbs along with their subject/objects, and leaving the identification of other adjuncts such as temporal phrases to downstream processes.

So we designed two additional features to model the argument consistency of the verbs x and y.

1. Subject consistency: whether the subjects of x and y (when present) are consistent;
2. Object consistency: whether the objects of x and y (when present) are consistent.

To characterize the consistency of the arguments (subjects and objects) between a hypothesis verb x and a premise verb y, here we developed a simple approach as a baseline. Take subject consistency for example, we let s_x be the subject term of verb x in the hypothesis, and let s_y be the aligned term of s_x in the premise (if there are multiple terms that are aligned with s_x, let s_y be the one that is closest to y in the dependency structure of the premise). The subject consistency of the verbs (x, y) is

then measured by the distance between s_y and y in the dependency structure of the premise.

The idea here is, if x and y are aligned, then for the subject of x, s_x, it's aligned part in the premise (s_y) should also be the subject of y. The distance between s_y and y characterizes (primitively) the possibility of s_y being y's subject.

Similarly, the object consistency of (x, y) is measured by the distance between the verb y and the aligned object of x.

An Example of Feature Estimation for Verb Alignment

Here we demonstrate how we estimate the features for verb alignment, using the example in Figure 3.2. Particularly, we show what are the feature values to decide the alignment between the hypothesis term $x_4 = reached$ and the premise term $y_7 = arrived$.

The values of primary features to decide this alignment are:

- String equality: 0
- Stemmed equality: 0
- WordNet similarity: 0.84
- Distributional similarity: 0.10
- Verb *be* identification: 1

Next we check the subject and object consistencies for the pair of verbs. Here we illustrate the object consistency as an example. We first find the object of x_4 in the hypothesis, $x_2 = San\ Francisco$. Assuming we have the result from the noun term alignment model that x_2 in the hypothesis is aligned to y_4 in the premise ($y_4 = San\ Francisco\ Bay$), we can then get the distance between y_4 and y_7 in the dependency structure of the premise (see Figure 3.2), which is 2 ($y_4 \sim y_6 \sim y_7$).

As such the argument consistency features for the verb pair (x_4, y_7) have values of:

- Subject consistency: 1 (the distance between y_7 and the aligned term of x_4's subject)
- Object consistency: 2 (the distance between y_7 and the aligned term of x_4's object, as illustrated above)

3.3.2 Features for the Inference Model

In Section 3.2.2 we introduced the inference model, which predicts the probability that a hypothesis clause s_j is entailed from a set of premise clauses $d_1 \dots d_m$, given a feature vector $\mathbf{f}_E$ describing these clauses with an alignment g between the terms in them:

$$\theta_E : \mathbf{f}_E(d_1, d_2, \dots, d_m, s_j, g) \rightarrow P(d_1 d_2 \dots d_m \vDash s_j)$$

We designed different feature sets according to whether s_j is a property clause or a relational clause.

Features for Property Inference Model

If s_j is a property clause, i.e., it takes one argument and can be denoted as $s_j(x)$, then for it to be inferred, we would like x's counterparts (i.e., aligned terms) in the premise to have the same or similar property.

Therefore, we look for all the property clauses in the premise that describe the counterparts of x, i.e. a clause set $D' = \{d_i(y) | d_i(y) \in D, g(x, y) = 1\}$. For example,

Premise: I've just heard some old songs. They're wonderful!
Hypothesis: I heard good music.

Consider the property clause $good(x_2)$ in the hypothesis with the term $x_2 = music$. Suppose that x_2 is aligned to two terms in the premise: $y_2 = songs$ and $y_4 = they$, then $D' = \{some(y_2), old(y_2), wonderful(y_4)\}$.

We then design a set of features to characterize the similarity between the clause s_j and the clauses in D'. These features are similar to those used in the alignment models in Section 3.3.1:

1. String equality: whether any of the clauses in D' is the same as s_j;
2. Stemmed equality: whether any of the clauses in D' has the same stem as s_j;
3. WordNet similarity: calculate the WordNet similarity (see Section 3.3.1 for definition) between any clause in D' and s_j, and pick the maximum one;
4. Distributional similarity: calculate the distributional similarity (see Section 3.3.1 for definition) between any clause in D' and s_j, and pick the maximum one.

In the above example, one property of y_4, $wonderful(y_4)$, has a high similarity to the property of $good(x_2)$, so we can predict that $good(x_2)$ is entailed from the premise.

Features for Relational Inference Model

If s_j is a relational clause, i.e., it takes two arguments and can be denoted as $s_j(x_1, x_2)$, then for it to be inferred, we would like the same or similar type of relation to exist in the premise, between x_1's and x_2's counterparts.

So we look for the sets of terms in the premise that are aligned with x_1 and x_2, respectively:

$$D'_1 = \{y | y \in D, g(x_1, y) = 1\}$$

$$D'_2 = \{y | y \in D, g(x_2, y) = 1\}$$

Table 3.1: Calculating the features of inference model for the example in Figure 3.2

Hypothesis clause	$subject(x_4, x_1)$		$object(x_4, x_2)$		$in(x_4, x_3)$	
Clause type	relational		relational		relational	
Terms in this clause	x_4	x_1	x_4	x_2	x_4	x_3
Aligned terms in the premise	$\{y_6, y_7\}$	$\{y_1\}$	$\{y_6, y_7\}$	$\{y_4\}$	$\{y_6, y_7\}$	$\{y_5\}$
Closest term pair in the premise	(y_7, y_1)		(y_6, y_4)		(y_6, y_5)	
Minimal distance f_r	1		1		1	

We then model the relations between the terms in D'_1 and the terms in D'_2. As a baseline approach, here we only develop one feature to model these relations. That is, the closest distance between these two sets of terms in the dependency structure of the premise:

$$f_r(D, s_j, g) = \min_{y_1 \in D'_1, y_2 \in D'_2} dist(y_1, y_2)$$

The idea here is simple: the closer that two terms are in a dependency structure, the more likely these two terms have a direct relationship. Since these relations are mostly syntactic relations (e.g., subject, object, etc.), we made an assumption that the closest relation found between D'_1 and D'_2 is the same type as the relation of s_j between x_1 and x_2.

An Example of Feature Estimation in Inference Model

We use the example in Figure 3.2 to illustrate how features are calculated for the inference model.

Suppose the alignment for this example is the one shown in Figure 3.3, then the inference features for each clause in the hypothesis are shown in Table 3.1.

3.4 Post Processing

According to Equation (3.1), when our inference model predicts that each of the clauses s_j in a hypothesis is entailed from the clauses $d_1 \dots d_m$ in a premise, the

whole hypothesis S is determined to be entailed from the premise D. However, this is not always true due to some of the linguistic phenomena, in particular, polarity and monotonicity. In our entailment system, we developed a post processing routine to deal with these issues.

3.4.1 Polarity Check

Consider the following example:

> **Premise**: Around this time, White decided that he would not accept the $10,000 Britannia Award and another Miles Franklin Award for his work.
>
> **Hypothesis**: White got the Britannia Award.

The hypothesis contains following terms and clauses:

> Terms: $x_1 = White$, $x_2 = Britannia\ Award$, $x_3 = got$
>
> Clauses: $subject(x_3, x_1)$, $object(x_3, x_2)$

When alignment between the hypothesis and the premise contains the following term pairs

$$(x_1, he),\ (x_2, Britannia\ Award),\ (x_3, accept)$$

all the clauses in the hypothesis can be inferred from the premise:

> $subject(x_3, x_1)$: x_1's aligned term (he) is the subject of x_3's aligned term ($accept$) in the premise.
>
> $object(x_3, x_2)$: x_2's aligned term ($Britannia\ Award$) is the object of x_3's aligned term ($accept$) in the premise.

However, in this example the entire hypothesis is clearly not entailed. This is because in the premise there is a negative adverb *not* applying on the verb *accept*.

Table 3.2: The list of negative modifiers used for polarity check

barely	nor
hardly	not
little	n't
neither	nowhere
never	rarely
no	scarcely
none	seldom

In order to detect this situation, the first post processing after predicting a positive entailment is to check the polarity of each verb in the hypothesis against its counterpart (i.e., aligned verbs) in the premise. If the polarities of a pair of aligned verbs are different, we change the entailment prediction to be false.

The polarity of a verb can be characterized by the number of its negative modifiers. A set of negative modifiers that we recognize are listed in Table 3.2.

3.4.2 Monotonicity Check

The monotonicity assumption states that, when a statement is true, adding any context would not affect the truth of that statement. This assumption, which may be true in the most studied formal logic, is however not the case in natural language. For example:

> **Premise**: He said that "there is evidence that Cristiani was involved in the murder of the six Jesuit priests" which occurred on 16 November in San Salvador.
>
> **Hypothesis**: Cristiani killed six Jesuits.

The hypothesis *Cristiani killed six Jesuits* can be sufficiently inferred from the statement *Cristiani was involved in the murder of the six Jesuit priests*. However, this example is a false entailment because the entailing statement is in a context of *he said that "..."*.

So after our system makes an positive entailment prediction, we also check against the monotonicity assumption. Our approach is to search the context of the entailing statement, namely, all the upward nodes from the head of that statement in the parse tree. If any of these nodes contain a non-monotonic context, the entailment prediction is changed to false.

From training data (see Section 3.5) we identified the types of words that signal non-monotonic text. They usually contain one of the following meanings:

1. Indicating a statement is someone's claim or declaration, e.g., *say*;
2. Indicating a statement is someone's vision or imagination, e.g., *think*;
3. Indicating a statement is someone's intended outcome, e.g., *suggest*;
4. Indicating a statement is questioned, e.g., *ask*;
5. Indicating a statement is hypothesized, e.g., *suppose*;
6. Indicating something is desired but may not actually happened, e.g., *prefer*;
7. Indicating something is permitted but may not actually done, e.g., *allow*;
8. Indicating something is weakly perceived but not attested or confirmed, e.g., *hear* (note that words expressing strong perception are considered to indicate true entailments, e.g., *witness*);
9. Indicating something is planned or happens in the future, e.g., *decide*;
10. Indicating something happened in the past, e.g., *use to*;
11. Indicating something is fake, e.g., *pretend*.

We further expanded this set of non-monotonic contexts by adding the synonyms of the recognized words. The expanded set of non-monotonic contexts are listed in Table 3.3.

Table 3.3: The list of non-monotonic contexts
(A extensive set also includes their derivative forms, e.g., *thought*)

advertise	dream	must	reckon
advise	elect	need	recommend
aim	enable	negotiate	report
allege	encourage	obligate	request
allow	enjoy	offer	require
announce	enunciate	opt	say
anticipate	envisage	order	seek
argue	expect	ought to	select
arrange	express	perhaps	shall
articulate	fancy	permit	solicit
ask	feel	phrase	speculate
assert	forecast	picture	state
assume	foresee	plan	suggest
attempt	foretell	plead	suppose
authorize	formulate	pose	surmise
beg	going to	possible	suspect
believe	guess	postulate	swear
call for	have to	potential	tell
can	hear	predict	tend
choose	hope	prefer	think
claim	hypothesize	premise	try
conceive	imagine	prepare	urge
conjecture	inquire	presume	use to
consider	insist	presuppose	vision
dare	intend	pretend	visualize
decide	let	probable	vote
declare	like	proffer	vow
deem	likely	project	want
demand	look forward to	promise	will
deserve	love	pronounce	wish
desire	may	propose	wonder
determine	maybe	propound	write
discuss	mean	put	
divine	might	question	

3.5 Experimental Results

We choose the textual entailment data from PASCAL-3 RTE Challenge [36] for our experiments. There are 800 entailment examples for the development set and 800 entailment examples for the test set. In order to train our entailment models, we first decomposed the premises and hypotheses in the development set into sets of terms and clauses, and then manually annotated the data set for ground-truth judgements described in Section 3.2, which are

- For each term in the hypothesis x and each term in the premise y, we annotated the label of $g(x, y)$ (whether x and y are aligned)[1];
- For each clause s_j in the hypothesis, we annotated whether it is entailed from the clauses $d_1 \dots d_m$ in the premise (truth value for $d_1 \dots d_m \models s_j$).

We then evaluated the results for both the alignment decision and the entailment prediction.

3.5.1 Alignment Results

We trained two logistic regression models from the annotated development data, one for noun term alignment and one for verb term alignment. Since we have only the gold-standard alignments for the development data (not for the test data), we evaluate their performances by cross-validation on the development data.

The evaluation is based on pairwise judgements: for a term pair (x, y), where x is from a hypothesis and y is from a premise, whether the model correctly predicts the value of their alignment function $g(x, y)$ (0 or 1). Since the class distribution of alignment judgement is extremely unbalanced (among all possible pairings of two

[1]The RTE-2 data set has word-level alignment annotation available [14], which is also an option to derive the ground truth for term-level alignments.

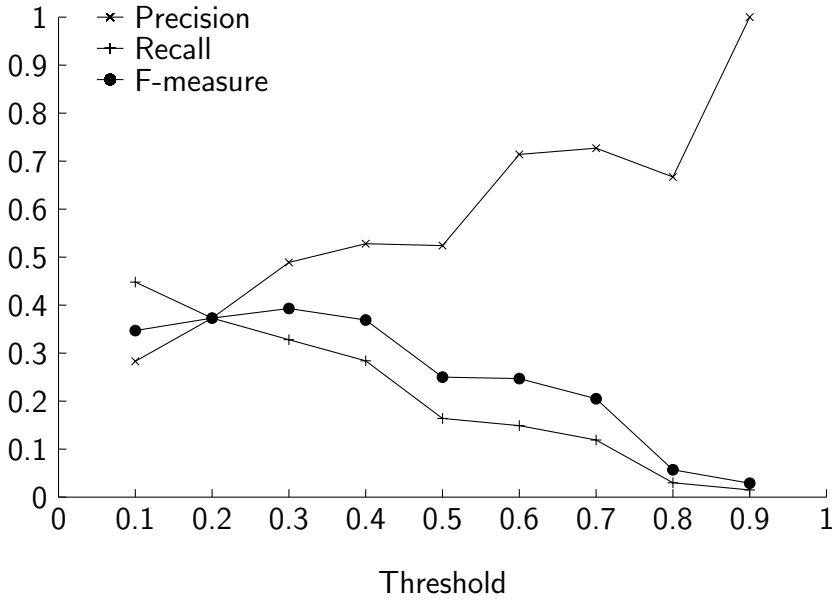


Figure 3.4: Evaluation results of verb alignment for textual entailment

terms, only a small portion of them are aligned pairs), we evaluate the alignment results by precision and recall of positive alignments.

The alignment for noun terms achieved 96.4% precision and 94.9% recall. This performance is relatively satisfying. We consider it sufficient for downstream processes.

The alignment for verb terms, however, performs significantly lower. The standard logistic regression model gives 52.4% precision and 16.4% recall. Since the recall performance is especially low, and it is actually more important to the downstream process (i.e., the inference model), efforts were made to balance the precision and recall. Our mechanism was to adjust the output threshold of the logistic regression model: the lower the threshold is, the model predicts more positive results (i.e., aligned term pairs), giving lower precision and higher recall; while the higher the threshold is, the model predicts more negative results (i.e., unaligned term pairs), giving higher precision but lower recall. We experimented with different thresholds from 0.1 to 0.9, and the results are shown in Figure 3.4.

We can see that the combined performance of precision and recall (i.e., the f-measure) achieved maximum when the threshold is set to 0.3. Under this setting the verb alignment model has a performance of 48.9% precision, 32.8% recall, and 39.3% f-measure.

3.5.2 Entailment Results

We then trained two logistic regression models for the inference model, namely, the property inference model and the relational inference model. The models were trained on the annotated examples of the development set, and applied to the test set. We evaluated the results predicted by these models. Among the 800 test examples, the entailment predictions made by our models achieved an accuracy of 60.6%. Comparing this result to the median performance of the participating systems in the PASCAL-3 RTE Challenge [36] (61.8%), the difference is not statistically significant (z-test, $p = 32\%$).

As discussed in Section 2.1.4, the key issue that distinguishes the performance of different systems is the amount of knowledge they use. In our implementation, we used knowledge sources and language tools no more than those publicly available, such as the Stanford parser [52], OpenNLP tools[2], WordNet [64], and the AQUAINT Corpus of English News Text [39]. Therefore, the fact that the performance of our implementation is on par with the median performance in RTE-3 provides a reasonable baseline to process conversation entailment.

[2] `http://opennlp.sourceforge.net/`

Chapter 4

An Initial Investigation on Conversation Entailment

As an initial investigation, we follow the PASCAL practice and created a database of examples on conversation entailment. We tested the dependency-based approach on the collected data. In this chapter, we describe our data collection and annotation procedure, analyze the collected data, and report the results from our initial investigation.

4.1 Problem Formulation

Following the PASCAL practice [8, 10, 22, 36, 37], here we consider the conversation entailment problem as inferring a single natural language statement, or a declarative sentence, from a conversation.

Similar to the formulation in Section 3.1, we use S to represent the statement which is the hypothesis in question, and use D to represent the premise from which the hypothesis is to be inferred. In this case the premise D is a conversation segment. We say that D **entails** S if and only if the meaning of S can be sufficient inferred

from the premise D, and write it as

$$D \models S$$

Similarly, if D does not entail S, we say

$$D \not\models S$$

Also similar to the case of textual entailment in PASCAL, the definition here is not strict. Rather, it is based on an agreement of most intelligent human readers, given the general background knowledge. That means, the standard is not whether the hypothesis is logically entailed from the premise, but whether it can be reasonably inferred by human readers.

Table 4.1 gives a few examples of premise-hypothesis pairs, and whether each hypothesis is entailed by the corresponding premise. These examples show that conversations are different from written text. Utterances in a conversation tend to be shorter, with disfluency, and sometimes incomplete or ungrammatical. These examples also show the importance to model the conversation context. One utterance could span several turns (e.g., utterance of B in Example 1). The pronouns are frequently used and may require special treatment (e.g., *you* in Example 2).

4.2 Types of Inference from Conversations

In the text entailment exercise, almost all hypotheses are about facts that can be inferred from the text segment. This is partly due to the fact that the newswire articles mainly report significant events and partly due to how the data is collected. From conversations, however, we can infer different types of information. It could be some opinion of the world held by the participants, some facts (assuming speakers

Table 4.1: Examples of premise-hypothesis pairs for conversation entailment

ID	Premise	Hypothesis	Entailed
1	B: My mother also was very very independent. She had her own, still had her own little house and still driving her own car, A: Yeah. B: at age eighty-three.	B is eighty-three.	False
		B's mother is eighty-three.	True
2	A: sometimes unexpected meetings or a client would come in and would want to see you, B: Right.	Sometimes a client wants to see B.	False
		Sometimes a client wants to see A.	True

are telling the truth) about the participants, and communicative relations between the participants (e.g., A disagrees with B).

In this work, we particularly focus on the inference about conversation participants. This is because understanding conversation participants is key to any application involving conversation processing: either acquiring information from human-human conversation or enabling human-machine conversation. In human-human conversation, correct hypotheses about conversation participants can benefit many applications such as information extraction and knowledge discovery from conversation data. In human-machine conversation, better understanding of its conversation partners will enable more intelligent system behavior.

Specifically, we are interested in following four types of inference:

- **Fact**. Facts about the participants. This includes:

    1. Profiling information about individual participants (e.g., occupation, birth place, etc.);
    2. Activities associated with individual participants (e.g., A bikes to work everyday);
    3. Social relations between participants (e.g., A and B are co-workers, A and B went to college together).

- **Belief**. Participants' beliefs and opinions about the physical world. Any statement about the physical world in fact is a belief of the speaker. Such statements are not about the speaker him/herself and often involve subjective judgements, e.g., *B thinks that crafts are relaxing.* Technically, the state of the physical world that involves the speaker him/herself is also a type of belief. However, here we assume a statement about oneself is true and is considered as a *fact*.

- **Desire**. Participants' desire of certain actions or outcomes (e.g., A wants to find a university job). These desires represent the states of the world the participant finds pleasant (although they could be conflicting to each other).

- **Intent**. Participants' deliberated intent, in particular communicative intention which captures the intent from one participant on the other participant such as whether A agrees/disagrees with B on some issue, whether A intends to convince B on something, etc.

Most of these types are motivated by the Belief-Desire-Intention (BDI) model [2], which represents key mental states and reflects the *thoughts* of a conversation participant. *Desire* is different from *intention*. The former arises subconsciously and the latter arise from rational deliberation that takes into consideration desires and beliefs [2]. The *fact* type represents the facts about a participant. Both thoughts and facts are critical to characterize a participant and thus important to serve many other downstream applications.

4.3 Data Preparation

Currently there is no data available to support the research on conversation entailment. Therefore, as a first step, we have developed a database of entailment examples

with different types of hypotheses to facilitate algorithmic development and evaluation.

4.3.1 Conversation Corpus

The data was collected from the Switchboard corpus [38]. It is a corpus of make-up phone calls, where the participants, who do not know each other, exchange ideas and discuss issues of interest. These conversations are casual and free-form compared to goal-driven conversations (e.g., conversation about how to install a computer program). Inference from this set of conversations can be more challenging since the goals/subgoals are not explicit and topic evolvement can be unpredictable.

All of the conversations in this corpus have been transcribed by human annotators. A portion of it has been annotated with syntactic structures, disfluency markers, and discourse markers as a part of Penn Treebank [61].

As we are mainly interested in semantic analysis and inferring information from the conversations, we work on the conversation transcripts directly.

4.3.2 Data Annotation

We selected 50 conversations from the Switchboard corpus. In each of these conversations, two participants discuss a topic of interest (e.g., sports activities, corporate culture, etc), and has a full annotation of syntactic structures, disfluency markers, and discourse markers. We chose the conversations with annotation because the available annotations will enable us to conduct systematic evaluations of developed techniques, for example, by comparing performance of inference based on annotated information versus automatically extracted information from conversation.

We had 15 volunteer annotators read the selected conversations, and created a total of 1096 entailment examples. Each example consists of a segment from the con-

versation, a hypothesis statement, and a truth value indicating whether the hypothesis can be inferred from the conversation segment, given the contextual information from the whole history of that conversation session. The following guidelines are followed during the creation of entailment examples:

- The number of examples is balanced between positive entailment examples and negative entailment ones. That is, roughly half of the hypotheses are entailed from the premise, and half of them are not.

- Special attention is given to negative entailment examples, since any arbitrary hypotheses that are completely irrelevant will not be entailed from the conversation. So in order not to make the prediction of false entailment too trivial, a special guideline is enforced to come up with "reasonable" negative examples: the hypotheses should have a major portion of words overlapping with the premise.

A recent study shows that for many NLP annotation tasks, the reliability of a small number of non-expert annotations is on par with that of an expert annotator [79]. It is also found that for tasks such as affection recognition, an average of four non-expert labels per item are capable of emulating expert-level label quality. Based on this finding, in our study the entailment judgement for each example was further independently annotated by four annotators (who were not the original contributors of the hypotheses). As a result, on average each entailment example (i.e., a pair of conversation segment and hypothesis) received five judgements, including the one given by the original annotator (i.e. creator of the hypothesis).

4.3.3 Data Statistics

In total we collected 1096 entailment examples from the annotators. In this section we will analyze the collected data and give some important statistics.

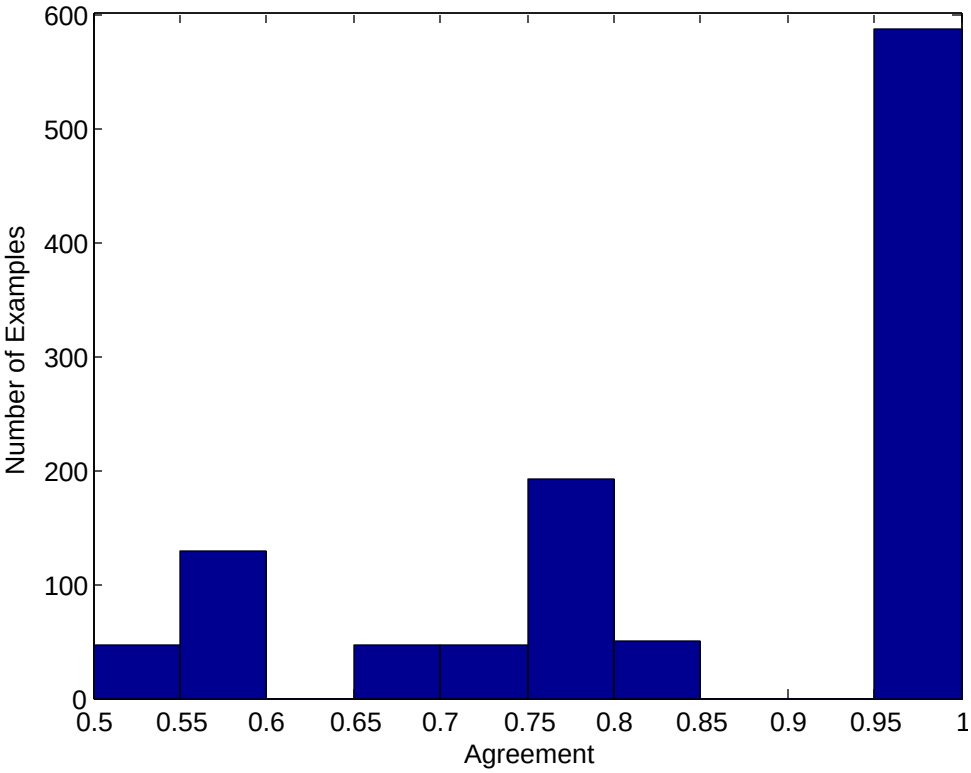


Figure 4.1: Agreement histogram of entailment judgements

As the most important annotation is the judgement of truth values, that whether a hypothesis can be inferred from the premise, it is essential to investigate how reliable those judgements are from our annotators, who are average native English speakers.

As described in Section 4.3.2, we have five entailment judgements from different annotators for each premise-hypothesis pair. Figure 4.1 gives a histogram of the agreements of collected judgements. From the figure we can see that inference from conversations is a difficult task, for only 53% of all the examples (586 out of 1096) are agreed by all human annotators.

Some of the disagreements are due to the ambiguity of the language itself, for example:

Premise:

A: Margaret Thatcher was prime minister, uh, uh, in India, so many,

uh, women are heads of state.

Hypothesis:

Margaret Thatcher was prime minister of India.

In the conversation utterance of speaker A, the prepositional phrase *in India* is ambiguous because it can either be attached to the preceding sentence, *Margaret Thatcher was prime minister*, which sufficiently entails the hypothesis, or it can be attached to the succeeding sentence, *so many women are heads of state*, which leaves it unclear which country Margaret Thatcher was prime minister of.

In some other instances of disagreements, the hypotheses are often not directly inferred from the text, but can be inferred after a few more steps of reasoning. Those reasonings often involve assumptions on conversational implicature or coherence. For example:

Premise:

A: Um, I had a friend who had fixed some, uh, chili, buffalo chili and, about a week before went to see the movie.

Hypothesis:

A ate some buffalo chili.

Premise:

B: Um, I've visited the Wyoming area. I'm not sure exactly where Dances With Wolves was filmed.

Hypothesis:

B thinks Dances With Wolves was filmed in Wyoming.

In the first example, a listener would assume that A follows the maxim of relevance, so that when she mentions the fixing of buffalo chili at this point in the conversation,

Table 4.2: Distribution of hypothesis types

	Count	Percentage
Fact	416	48.3%
Belief	299	34.7%
Desire	54	6.3%
Intent	92	10.7%

it is relevant. A most natural inference that would make the fixing of buffalo chili relevant is that A ate the buffalo chili.

In the second example, when the speaker A mentions a visit to the Wyoming area and expresses a lack of knowledge of the filming place of *Dances With Wolves*, the entire utterance is assumed to be coherent. This means in the speaker's mind, the Wyoming area must have some relationship with the filming of *Dances With Wolves*, although she does not know where exactly in the Wyoming area that movie was filmed.

Given the fact that the inference from conversations is already so difficult even for human readers, it is expected to be much more challenging for computer systems. Therefore for the first step we will focus our preliminary experiments on 875 entailment examples that have agreements greater than or equal to 75%.

For the 875 entailment examples that have good agreements ($\geq 75\%$), we observe a slight imbalance between the positive entailment class and the negative entailment class. The ratio is 474:401 (54%:46%), with a bias toward the positive class.

This also sets up a natural baseline for our entailment prediction system, as a majority guess approach (i.e. always guess positive for a data set that is biased to the positive class) will achieve 54% prediction accuracy, expectedly.

The distributions of four hypotheses types among the 875 data set are shown in Table 4.2.

4.4 Experimental Results

We applied the same dependency approach (as in Chapter 3) to the conversation entailment data. This section presents our preliminary experiments and initial findings.

4.4.1 Experiment Setup

As described in Section 4.3, our data set of conversation entailment consists of 875 premise-hypothesis pairs created from 50 conversations. To facilitate follow-up investigations, we further divided the 875 examples into two sets: a development set and a test set. We select one third of the examples as development data and two third as test data. The division is governed by the following guidelines:

1. No instances from the same conversation are divided into two different sets, since we will potentially train our computational models from the development data and apply them on the test data;

2. The ratio between positive and negative instances should remain roughly the same for both the development and the test data sets;

3. The distribution of four hypothesis types (fact, belief, desire, intent) should remain roughly the same on both the development and the test data sets.

As a result, we selected 291 examples from 15 conversations as the development set and 584 examples from 35 conversations as the test set. The positive/negative ratio and the distribution of hypothesis types in both data sets are presented in Table 4.3.

Similar to the discussion in Section 4.3.3, the natural baseline by always guessing the majority class can achieve accuracies of 56.4% and 53.1% on the development and test data sets, respectively.

Table 4.3: The split of development and test data for conversation entailment

	Total	Development	Test
Conversations	50	15	35
Premise-hypothesis pairs	875	291	584
Positive entailments	54.2%	56.4%	53.1%
Negative entailments	45.8%	43.6%	46.9%
Fact hypotheses	48.3%	47.1%	48.9%
Belief hypotheses	34.7%	34.0%	35.1%
Desire hypotheses	6.3%	10.7%	4.0%
Intent hypotheses	10.7%	8.2%	11.9%

4.4.2 Results on Verb Alignment

As shown in Section 3.5.1, the alignment for noun terms is a relatively easy task, for which our current model can already be considered sufficient and giving satisfying results. Thus in the follow-up evaluations for alignment models, we will focus on the alignment results for verb terms.

We applied the alignment model learned from the textual entailment data directly to the conversation entailment data. The first two series in Figure 4.2 (*Development* and *Test*) show the f-measures of the alignment results on the development set and the test set, respectively (here *Development* is only the name of a data set, which is not used to develop our system models just yet).

Similar to Section 3.5.1, here we also evaluate a series of results with different output thresholds for the logistic regression model. We can see that both f-measures for the development set and for the test set achieves to maximum when the threshold is set to 0.7 (24.9% and 32.3%, respectively). However, as we also show the f-measures in the textual entailment task as the third series (*Text*) in Figure 4.2, we can see that the maximum performance of conversation alignment is significantly lower than the maximum performance of text alignment (39.3%). This shows that the alignment model learned from textual entailment is not sufficient to tackle conversation entailment.

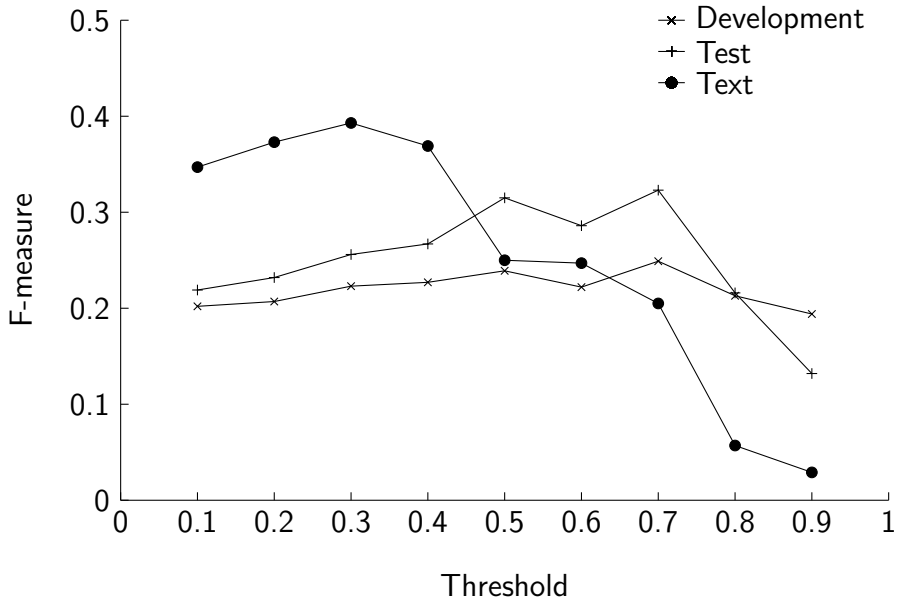


Figure 4.2: Evaluation results of verb alignment using the model trained from text data

4.4.3 Verb Alignment for Different Types of Hypotheses

We broke down the evaluation on verb alignment (with threshold 0.7) by different hypothesis types, and the results are shown in Figure 4.3. For both the development and the test data sets, the performance is better for the *fact* type than for most other types. For *fact* type of hypotheses, the alignment f-measures are consistent between the development and test data (30.7% and 37.3% respectively), which are also close to that on the text data (39.3%). However, the f-measures for other types of hypotheses are not so consistent, especially for *desire* and *intent* types. This is because there are not many instances in these two subsets of data. Nevertheless, if we combine the results on the development and test data for these two subsets, we get f-measures of 31.0% for *desire* and 27.0% for *intent*.

In summary, when we apply the alignment model learned from textual entailment to the task of conversation entailment, it handles the alignments for *fact* and *desire* types of hypotheses with relatively acceptable performance. The limitation of

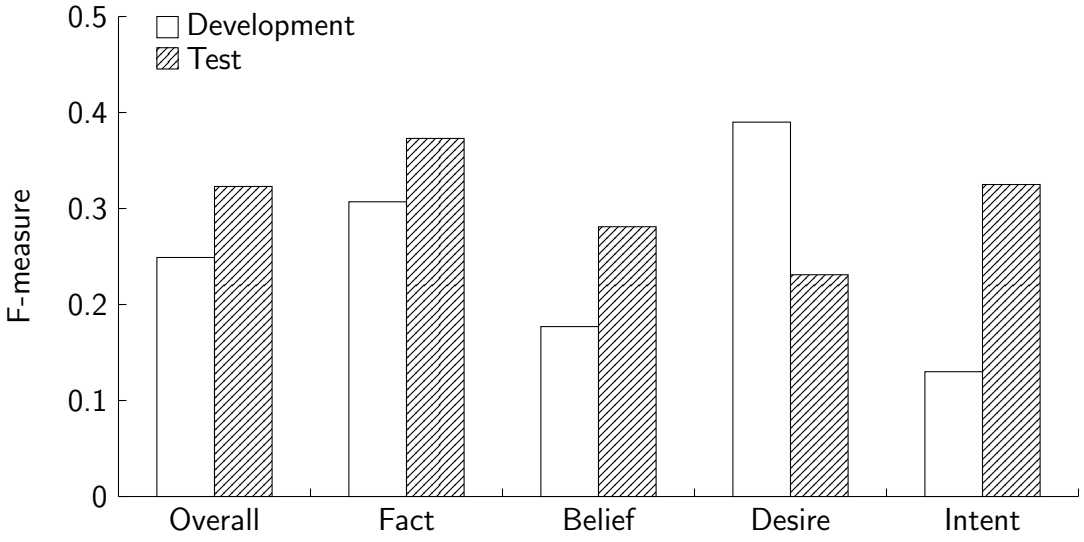


Figure 4.3: Evaluation results of verb alignment for different types of hypotheses

the current model is mostly revealed when dealing with *belief* and *intent* types of hypotheses.

4.4.4 Results on Entailment Prediction

Again, we applied the inference models learned from the textual entailment data directly to the conversation entailment data. Figure 4.4 shows the performances for both the development set and the test set. The overall prediction accuracies for the two data sets are 48.5% and 53.1%, respectively. Similar to what we found from the alignment evaluation, the reasonable models for predicting textual entailment now produces significantly lower performance on the conversation data.

In fact, the performance of the model predictions did not even beat the baseline of majority guess, which (as given in Section 4.3.3) are 56.4% for the development set and 53.1% for the test set. This is probably because our approach takes a rather strict standard, i.e., it tends to predict negative entailments rather than positive entailments. As a result, the more a data set is biased towards positive class (e.g., the development set), the less accurately our approach performs.

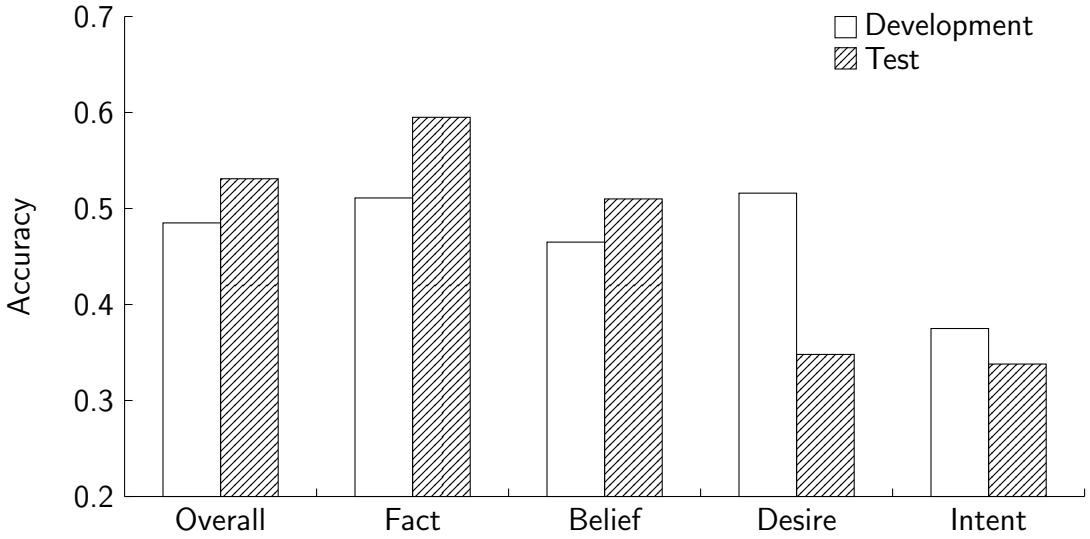


Figure 4.4: Evaluation results of entailment prediction using models trained from text data

Could the performance difference be attributed to the different sources of training data? We further experimented with training our entailment models (both alignment and inference models, with the same set of features as in textual entailment case) from the development data of conversation entailment, and evaluating them on the test data. This resulted in an accuracy of 52.4%. The newly trained models show no advantage compared to the previous models trained from textual entailment data. Therefore in the follow-up investigations, we will still use the previous result (53.1% accuracy on the test data) as the baseline.

Figure 4.4 also shows the break-down results of entailment performances by different hypothesis types. Again we see the current models perform better for *fact* type than the other three types.

The initial results on conversation entailment suggest that only applying approaches from textual entailment will not be enough to handle entailment from conversations, especially the entailment of *belief*, *desire*, and *intent* types of hypotheses. Considerations of unique behaviors of conversations is important to tackle the conversation entailment problem.

Chapter 5

Incorporating Dialogue Features in Conversation Entailment

Dialogues exhibit very different language phenomena compared to monologue text. As a result, the algorithm framework that is designed to recognize entailment from text will not be sufficient to process conversation entailment. In order to effectively predict entailment from conversations, we need to model unique features from the conversations [92]. In this chapter we discuss the modeling of two types of features: linguistic features in conversation utterances and structural features of conversation discourse.

5.1 Linguistic Features in Conversation Utterances

Compared to newswire texts that are mostly formal and standardized, spontaneous conversations tend to have much more linguistic variations, which dramatically increases the difficulty of recognizing entailments from them. These variations of linguistic features mainly include disfluency, syntactic variations, and special usage of language.

5.1.1 Disfluency

Oral conversations contain different forms of disfluency that breaks the normal structure of language. Below are a list of some types of disfluency.

1. Filled pause

   When people are thinking, hesitating, or pausing their conversation due to other reasons while they speak, they tend to create such words like *uh*, *um*, *huh*, etc. These insertions have no semantic content, but they break the flow of communication.

2. Explicit editing term

   These are the words that have some semantic content, but do not carry much actual meaning, such as *I mean*, *sorry*, *excuse me*, etc. For example:

   > A: Oh, yeah, uh, the whole thing was small and, you, *I mean*, you actually put it on.

   They usually occur between the restart and the repair (and are as such "explicit").

3. Discourse marker

   Discourse markers does not carry much meaning either, but they have a wider distribution than explicit editing terms. Such words include *like*, *so*, *actually*, etc. For example:

   > B: I think that was better than *like* Showbiz Pizza cause there's more of them to do.

   Because discourse markers can almost appear anywhere in a sentence, they are much more likely to be confounded with content words that take the exact same

form, and thus create ambiguities. Take the word *like* for example and compare its roles in sentences *They're like bermuda shorts* and *They're, like, bermuda shorts*. In the first case the shorts are not bermudas (only look like them), while in the second case they are.

4. Coordinating conjunction

   These are conjunctions like *and*, *and then*, *and so*, etc. But unlike regular conjunctions, they carry no semantic meanings while just serve as coordinating roles. Example:

   A: *And* he usually is good about staying within them, although our next door neighbors have a dog, too, *and*, uh, she, she is good friends with my dog.
   B: Oh, yeah?
   A: *And so* he often gets to smelling her scent and will go over there to sniff around and stuff.

5. Aside

   Aside is a longer sequence of words that is irrelevant to the meaning of the main sentence. It interrupts the fluent flow of the sentence and the sentence later picks up from where it left off. For example:

   B: I, uh, talked about how a lot of the problems they have to come, overcome to, *uh, it's a very complex, uh, situation,* to go into space.

6. Turn interruption

   The speech of a speaker can be interrupted in the middle by another person and then continued and completed by the same speaker. For example:

B: I thought it was kind of a strange topic about corruption in the government and –

A: Yeah.

B: – uh, how many people are self serving.

7. Incomplete sentence

   Sometimes a sentence is incomplete. This may be because it is interrupted by another speaker and then discontinued, or it is just unfinished by the speaker. For example:

   A: *We've had him for,* let's see, he just had his fourth birthday.

8. Restart

   A restart happens when a part of a sentence is canceled by the speaker and then fixed by a repairing part. Examples of restarts can be a simple substitution:

   A: Show me flights *from Boston on, uh, from Denver on* Monday.

   or more complicated cases where there is a restart within a restart (which are called nested restarts):

   A: *I liked, uh, I, I liked* it a lot.

5.1.2 Syntactic Variation

Oral conversations have unique syntactic behavior which rarely occurs in written newswire articles. We summarize a few phenomena as follows.

1. Dislocation and movement

A dislocation describes the case when a sentence constituent (which is dislocated) is associated with a resumptive pronoun. For example, in

A: *John*, I like *him* a lot.

John is associated with *him*, which constitutes a left-dislocation. And in

A: One of the problems *they*'re facing now, *a lot of people* now, is that the small business can't offer health insurance.

a lot of people is associated with *they*, which constitutes a right-dislocation.

A similar case is the movement of appositives. While it is very much like the dislocation, the only difference is that the moved appositive is associated with a regular noun phrase other than a pronoun. For example:

B: *Her father* was murdered, *her father and three other guys up here in Sherman.*

In both of these cases, it is critical to recognize the dislocated or moved constituent and identify the original element they are associated with.

2. Subjectless sentence

In strictly grammatical sentences, those without subjects may in most cases be considered imperatives. In conversations, however, the use of empty subjects is allowed in non-imperative contexts. For example:

B: You know, I think you are right. I think it is Raleigh.
A: *Think so?*

In this example, a completed form of the sentence of speaker *A* should be *Do you think so?*.

Here is another example:

> B: Later I tried to get the baby to a baby-sitter. *Supposed to be good, recommended person from the church*, and I knew her personally.

for which an unabbreviated form would be *She was supposed to be good.*

In order to get the actual meaning of a subjectless sentence, a way to recover what has been omitted is desired.

3. Double construction

   Double constructions are rarely seen in written, textual English, and are thus in need of special treatment for both syntactic parsing and semantic interpretation. These include double *is* construction, such as in

   > B: That*'s* the only reason I work there, *is* that my children now have graduated, and graduated from college.

   and double *that* construction, for example:

   > A: Or you can hope *that* if people keep their money *that* they'll spend more and create jobs and, and whatnot.

5.1.3 Special Usage of Language

Compared to written text, language use in oral conversations can be much more flexible. Such flexibility can have significant influence on the recognition of conversation entailment. Below are a few situations of the special usages.

1. Ellipsis

   Ellipsis can happen in written text, but it is used much more widely and frequently in oral English. For example:

   A: Did, did you go to college?
   B: Well, no. I'm going right now.

   In this conversation, speaker B's utterance means *I'm going to college right now*. It is important to recognize such an ellipsis in order to recognize entailments like *B is going to college*.

   In Section 5.1.2, "subjectless sentence" is a special case of ellipsis. In that case a regular grammatical component of a sentence is omitted, making up a special syntactic structure. While here we consider sentences, although with ellipses, but still in ordinary syntax.

2. Etcetera

   There are many possible ways to represent etcetera in English, such as *and so on*, *and so forth*, etc. More variations are specifically seen in spoken English, including *or whatever*, *or something like that*, and *and stuff like that*. These vague phrases, which can be either nominal or adverbial, require special recognition to be distinguished from regular nominal or adverbial phrases. For example, a nominal etcetera can be used in conjunction with an enumeration of verb phrases:

   A: They just watch them and let them play *and things like that*.

3. Negation

In Section 3.4.1 we discussed the importance of modeling negation in textual entailment task. We have listed a set of negative adverbs in Table 3.2. However, negation in conversations can be represented by a larger variety of forms. For example:

> B: They've got to quit worrying about, uh, the, uh, religious, uh, overtones in our textbooks and get on with teaching the product.

In this utterance, the word *quit* also represents a meaning of negation, which is the same as saying *they've got to not worry about ...*.

4. Question form

   Written text also take question forms from time to time, but they are mostly rhetorical questions or hypothetical questions. In conversations, however, as two or more people communicate and exchange ideas and information, it is much more common to see one speaker ask a question, which is answer by another speaker. For example:

> B: Hi, um, okay what, now, uh, what particularly, particularly what kind of music do you like?
> A: Well, I mostly listen to popular music.

5.2 Modeling Linguistic Features in Conversation Utterances

As a starting step, we chose to incorporate the disfluency and some of the special usages of language in our conversation entailment system. This section describes how we model these features.

5.2.1 Modeling Disfluency

The detection of disfluency has been studied in previous works [70]. Here our focus is how they affect the recognition of conversation entailment, and how to model them in the entailment prediction process. Therefore, we employ a corpus of disfluency annotations on the Switchboard conversations, given by Penn Treebank [61].

After they are detected (or marked out by annotation), we treat different types of disfluency differently. Filled pauses, explicit editing terms, discourse markers, coordinating conjunctions, and asides are directly removed. Interrupted utterances are pieced together to recover the meaning of the original utterances. Incomplete sentences are ungrammatical, usually unable to analyze or comprehend, and often make no sense. Thus they are discarded from the conversations.

A more complex case is the restarts. They need to be repaired for their original meaning to be understood. For example,

A: Show me flights *from Boston on, uh, from Denver on* Monday.

In this case, we remove the canceled part (e.g., *from Boston on*) as well as concurrent filled pauses and editing terms (e.g., *uh*), and replace them with the fixed constituent (e.g., *from Denver on*). As such we are able to recover the correct form of this utterance: *Show me flights from Denver on Monday.*

5.2.2 Modeling Polarity

In Section 5.1.3 we have found a group of words that can represent negative polarities in conversations, which were not used in textual entailment. We expanded this group of words and added them to the set of negative modifiers used in textual entailment (in Table 3.2). The expanded set of negative words is listed in Table 5.1.

Table 5.1: The expanded set of negative words used for polarity check

The set used in textual entailment			
barely	never	not	scarcely
hardly	no	n't	seldom
little	none	nowhere	
neither	nor	rarely	
The expanded set			
abolish	deny	give up	proscribe
abort	disallow	halt	put off
abrogate	disapprove	hesitate	quit
annul	disclaim	interdict	refuse
avert	discontinue	invalidate	reject
avoid	drop	negate	repeal
ban	eliminate	neglect	repudiate
bar	escape	nullify	rescind
cancel	except	obviate	resist
cease	exclude	omit	revoke
debar	fail	oppose	stop
decline	forbid	postpone	terminate
defer	forestall	preclude	void
defy	forget	prevent	
delay	gainsay	prohibit	

Using the negative identifiers in Table 5.1 and the post processing mechanism of polarity check described in Section 3.4.1, we are now able to detect that conversation entailment examples such as

Premise:

B: They've got to quit worrying about, uh, the, uh, religious, uh, overtones in our textbooks and get on with teaching the product.

Hypothesis:

B believes people should focus more on the religious overtones of textbooks.

are false entailments.

5.2.3 Modeling Non-monotonic Context

In Section 3.4.2 we have proposed that after each entailment prediction, if it is predicted to be true entailment, we need to check the context of the entailing statement against the monotonicity assumption. In conversation entailment, we follow the same idea. However, the category of non-monotonic context should not be limited to what was introduced in Section 3.4.2. For example:

Premise:

B: What kind of music do you like?

Hypothesis:

A likes music.

The clause representation of the hypothesis is

$$x_1 = A,\ x_2 = music,\ x_3 = likes,\ subject(x_3, x_1),\ object(x_3, x_2)$$

In the conversation segment, we can align the term $y_1 = music$ to the hypothesis term $x_2(music)$, term $y_2 = you$ to the hypothesis term $x_1(A)$, and term $y_3 = like$ to the hypothesis term $x_3(likes)$. Since y_2 is y_3's subject in the premise, which entails the hypothesis clause $subject(x_3, x_1)$, and y_1 is y_3's object in the premise, which entails the hypothesis clause $object(x_3, x_2)$, all clauses in the hypothesis are entailed. According to the entailment framework in Section 3.1, the hypothesis can be predicted to be entailed from the conversation segment.

However, the hypothesis in this example is clearly not entailed from the premise, because the conversation segment provides no descriptive information about speaker A. In fact, the premise relations $subject(y_3, y_2)$ and $object(y_3, y_1)$, from which the hypothesis clauses are entailed, all occurred in a question asked by the speaker B.

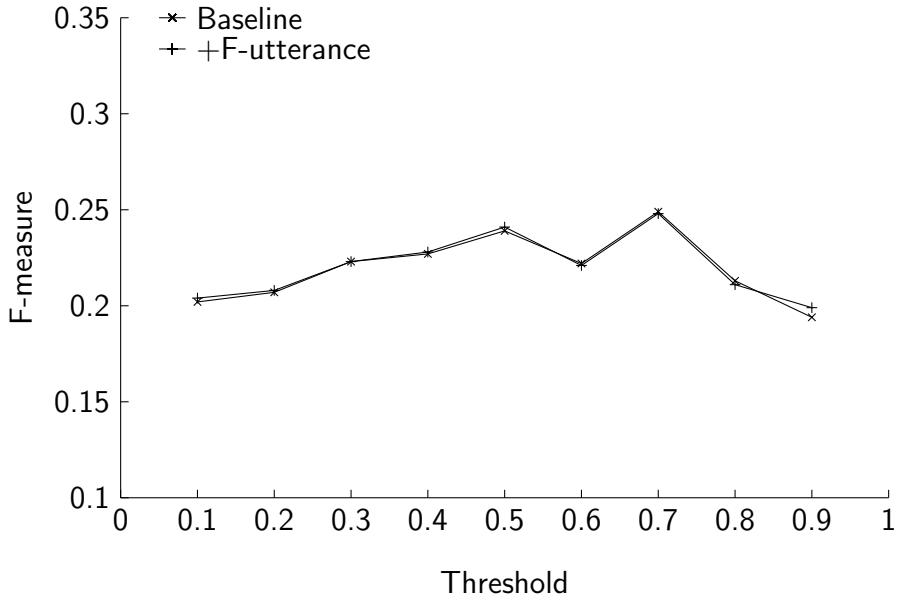
Therefore in conversation entailment, we identify questions (including *wh*-questions and *yes-no*-questions) as non-monotonic context too. Admittedly, questions can also be identified as non-monotonic context in textual entailment. But as we discussed in Section 5.1.3, question forms are not extensively used in text writing, while are much more common in conversations.

5.2.4 Evaluation

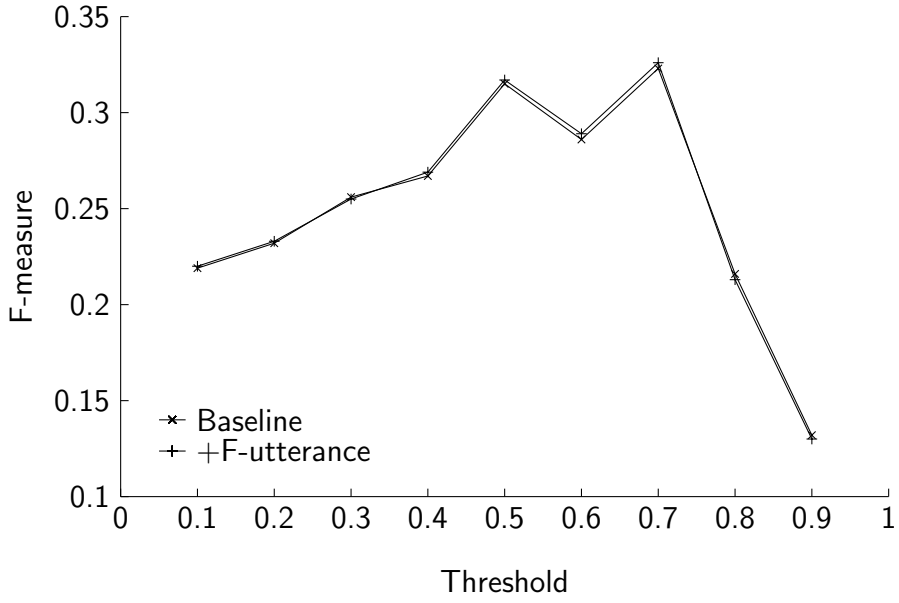
In order to evaluate our improved system modeling the linguistic features in conversation utterances, and compare it to the baseline system using models from textual entailment, we investigated both how they perform on the verb alignment task and how they classify entailment prediction.

Evaluation on Verb Alignment

Figure 5.1(a) and 5.1(b) show the verb alignment results on the development and the test data sets respectively. The *Baseline* results were produced by the system using models from textual entailment, and the *+F-utterance* are the results incorporating



(a) On development data



(b) On test data

Figure 5.1: Evaluation of verb alignment for system modeling linguistic features in conversation utterances

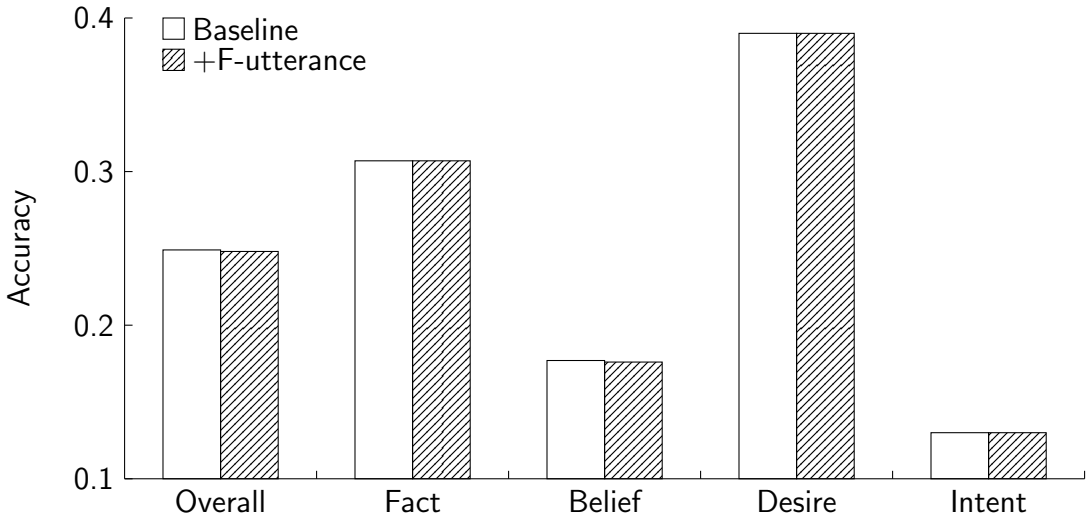
linguistic features in conversation utterances. Results are presented with different thresholds of model output.

As we see from the comparison, the two systems do not produce very much different results on verb alignment. This is not surprising, since the modeling of linguistic features (as described in Section 5.2) mostly happens on the post processing stage (polarity check or monotonicity check). Figure 5.2 of the alignment results broken down by hypothesis types again demonstrates similar comparisons between the two systems.

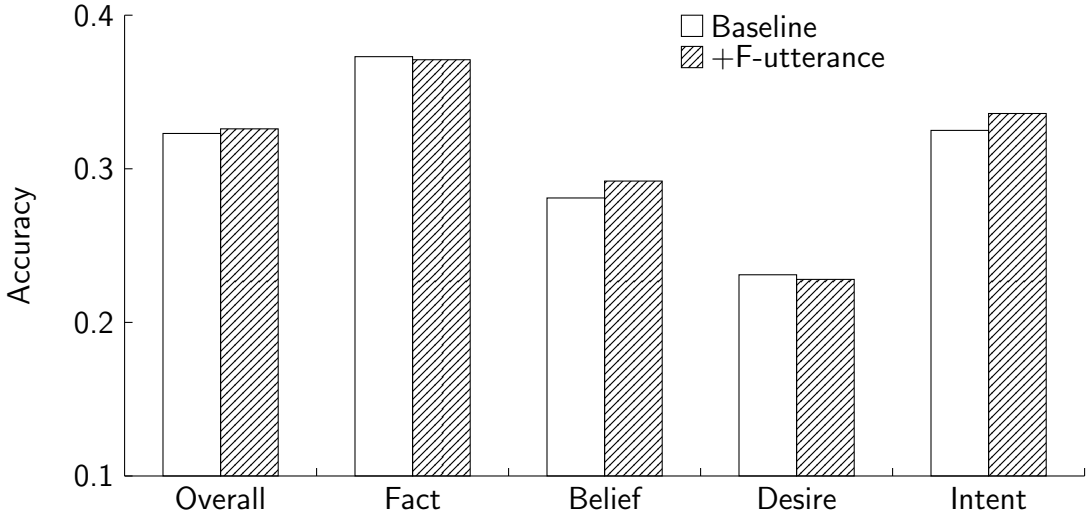
Evaluation on Entailment Prediction

Figure 5.3 compares the entailment prediction performances of the two systems for both the development and the test data sets. Overall speaking, the system incorporating linguistic features in conversation utterances (*+F-utterance*) shows some improvement over the *Baseline* system on the development data, but not much improvement on the test data. This is because the baseline on the development data is relatively low (as discussed in Section 4.4.4). Overall speaking, neither of the performances on development data and on test data has beaten the natural baselines by majority guess (56.4% and 53.1% respectively, as in Section 4.3.3) after modeling linguistic features in conversation utterances.

However, if we break down the evaluation results by different types of hypotheses, which are also shown in Figure 5.3, we can see that modeling linguistic features bring improvement on both data sets for the inference of *fact* type of hypotheses. Statistical tests illustrate that the improvements are significant on both data sets (McNemar's test, $p < 0.05$). This demonstrates that the modeling of linguistic features in conversation utterances helps identifying the entailment of *fact* hypotheses, but is not so effective for other types of hypotheses (*belief*, *desire*, and *intent*). The entailment of *belief*, *desire*, and *intent* hypotheses requires further modeling beyond

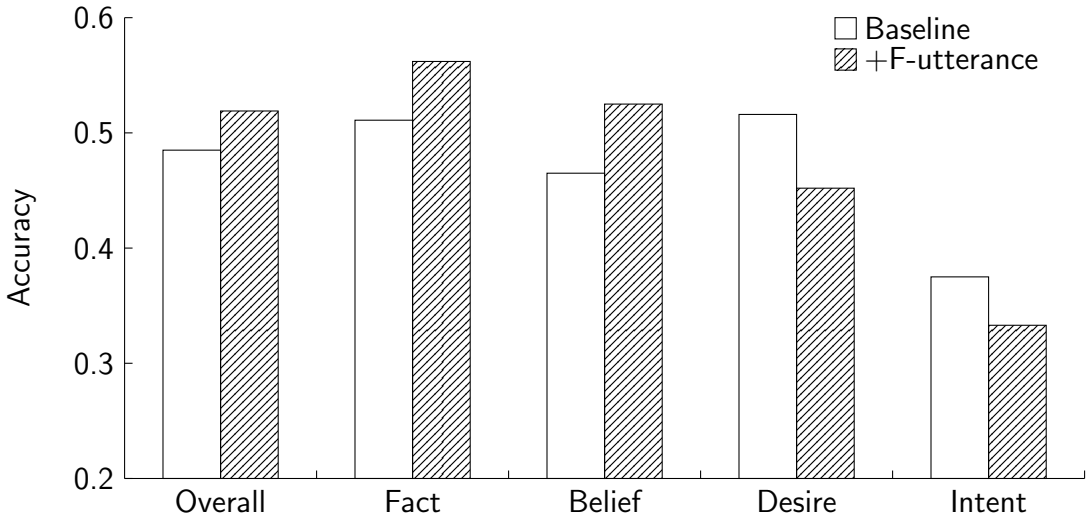


(a) On development data

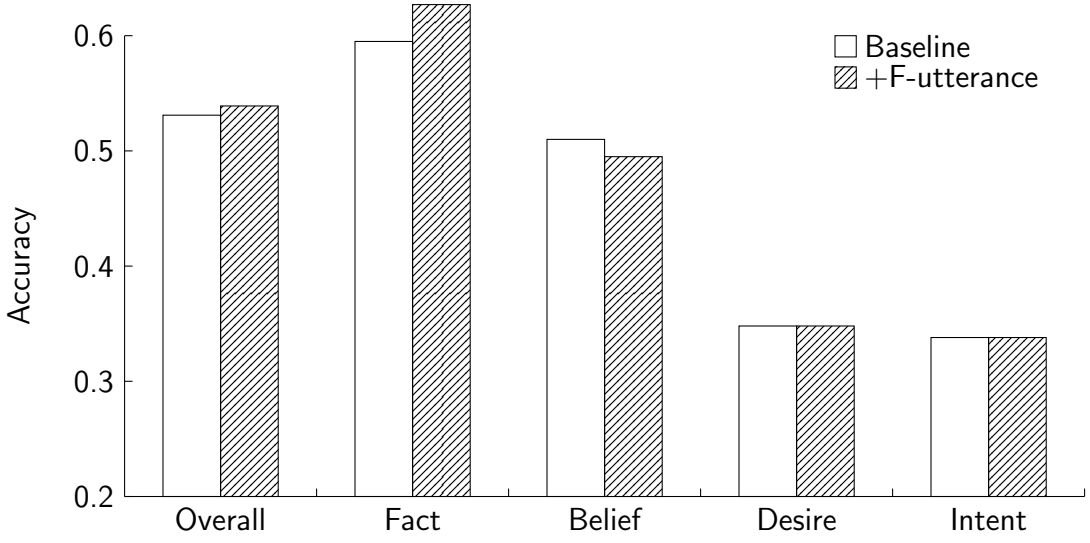


(b) On test data

Figure 5.2: Evaluation of verb alignment by different hypothesis types for system modeling linguistic features in conversation utterances



(a) On development data



(b) On test data

Figure 5.3: Evaluation of entailment prediction for system modeling linguistic features in conversation utterances

the conversation utterances, such as features of conversation structure.

5.3 Features of Conversation Structure

A more important characteristic of conversations is that they are communications between two or more people. Thus they contain interactions between the conversation participants, such as turn-taking, grounding, etc. These unique conversation features make the task of conversation entailment even more distinctive from that of textual entailment.

Consequently, modeling the structure of conversation interaction can be critical to recognizing conversation entailment. Below are several examples of conversation structures that need to be modeled in order to predict correct entailments.

1. Question and answer

   In the example,

   Premise:
   A: And where abouts were you born?
   B: Up on Person County.
   Hypothesis:
   B was born in Person County.

   speaker A asks a question and B gives an answer. In order to correctly infer the statement in the hypothesis, we need to consider B's answer under the context of A's question, i.e., that *up on Person County* in B's answer are adjuncts to the verb *born* in A's question.

   Generally for a *wh*-question and its answer, we need to identify the semantic relation between proper constituents from the question and from the answer respectively, so that a similar relation in the hypothesis can be entailed. For

a *yes-no*-question, however, we usually can find all desired relations from the question, and use the *yes* or *no* answer to validate or invalidate such relations. For example,

Premise:

A: Do they, were, were there, um, are you allowed to, um, be casual, like if it was summer, were, were you allowed to wear sandals, and those, tha-, not real,

B: Not really, I think, I mean it's kind of unwritten, but I think we're supposed to wear hose and, and shoes,

Hypothesis:

B is allowed to wear sandals.

In this case the hypothesis is not entailed because in the conversation segment speaker B gives a *no*-answer, which invalidates the relation between *you* and *allowed* in A's question.

2. Viewpoint and agreement

In the following example,

Premise:

A: We did aerobics together for about a month and a half and that went over real well,

B: Uh-huh.

A: but, uh, that's about it there.

B: Oh, it's good and it's healthy, too.

A: Oh, yeah, yeah.

Hypothesis:

A agrees with B that aerobics is healthy.

speaker B raises a viewpoint and speaker A agrees with it. There are three parts in the hypothesis related to the verb *agree*, the person that agrees (A), the

person that is agreed with (B), and the content that is agreed on (*that aerobics is healthy*). The content part can be entailed from speaker B's utterance (*it's good and it's healthy*) from the conversation segment, while the other two parts have to be inferred from the relation between utterance *Oh, yeah, yeah* and its speaker, and the relation between these two utterances.

A similar case is a viewpoint and a disagreement. For example,

Premise:

A: Of course, there's not a whole lot of market for seventy-eight RPM records.

B: Is there not? You, you'd, well you'd think there would be.

Hypothesis:

B disagrees with A about the market for seventy-eight RPM records.

3. Proposal and acceptance

In the example below,

Premise:

B: Have you seen *Sleeping with the Enemy*?

A: No. I've heard that's really great, though.

B: You have to go see that one.

A: Sure.

Hypothesis:

A is going to see *Sleeping with the Enemy*.

speaker B makes a proposal and speaker A accepts it. Again here we need to consider speaker B's utterance *you have to go see that one* and speaker A's utterance *sure* together to predict the entailment of the hypothesis statement. Terms and relations in the hypothesis can be entailed by the terms and relations

in B's utterance, but the whole statement has to be validated by A's acceptance of this proposal.

Similarly, there can be proposals and denials, which also requires the modeling of conversation structure to be correctly recognized.

5.4 Modeling Structural Features of Conversations

In order to model the structural features of conversations in our entailment system, we first incorporate the conversation structures in our representation of conversation segments, i.e., the clause representations.

5.4.1 Modeling Conversation Structure in Clause Representation

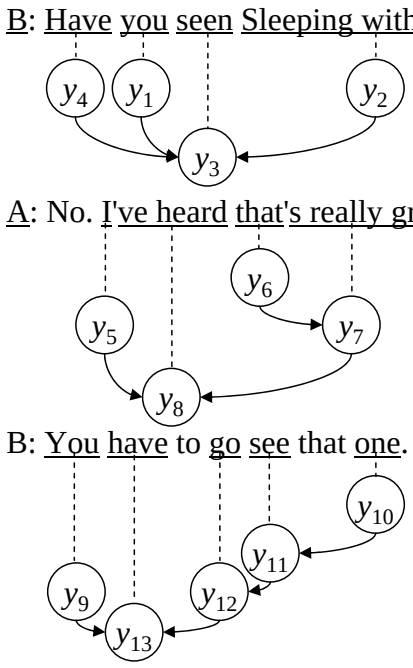
Previously we have used the same technique to represent the utterances in conversations as we used in representing text. For example, Figure 5.4(a) shows an example of a conversation segment (with a corresponding hypothesis), and Figure 5.4(c) shows the clause representation for the conversation segment in this example. As we described in Section 3.1.1, a clause representation is equivalent to dependency structure. So Figure 5.4(b) also shows the dependency structure of the conversation utterances.

This representation represents only the information in conversation utterances, without the information of the conversation structure. In order to incorporate structural features such as speaker identity, turning, and dialogue acts, we propose to augment the representation of conversation segments by introducing additional terms and predicates:

- **Utterance terms**: we use a group of pseudo terms u_1, u_2, ... to represent in-

Premise:
B: Have you seen *Sleeping with the Enemy*?
A: No. I've heard that's really great, though.
B: You have to go see that one.
Hypothesis:
B suggests A to watch *Sleeping with the Enemy.*

(a) a conversation segment (premise) and a corresponding hypothesis



(b) dependency structure of the conversation utterances

Terms	Clauses
y_1=*A* y_2=*Sleeping with the Enemy* y_3=*seen*, y_4=*have*	*obj*(y_3,y_2) *subj*(y_3,y_1) *aux*(y_3,y_4)
y_5=*A*, y_6=*that* y_7=*is really great* y_8=*have heard*	*subj*(y_7,y_6) *obj*(y_8,x_7) *subj*(y_8,y_5)
y_9=*A*, y_{10}=*one*, y_{11}=*see*, y_{12}=*go*, y_{13}=*have*	*obj*(y_{11},y_{10}) *obj*(y_{12},y_{11}) *obj*(y_{13},y_{12}) *subj*(y_{13},y_9)

(c) clause representation of the conversation utterances

Figure 5.4: An example of dependency structure and clause representation of conversation utterances

dividual utterances in the conversation segment. We associate the dialogue acts for each utterance with the corresponding terms, e.g., $u_1 = yes\_no\_question$. Here we use a set of 42 dialogue acts from the Switchboard annotation [38]. Appendix B lists the dialogue act set.

- **Additional speaker terms**: we use two terms s_A, s_B to represent individual speakers in the conversation. These terms can potentially increase for multi-party conversations.

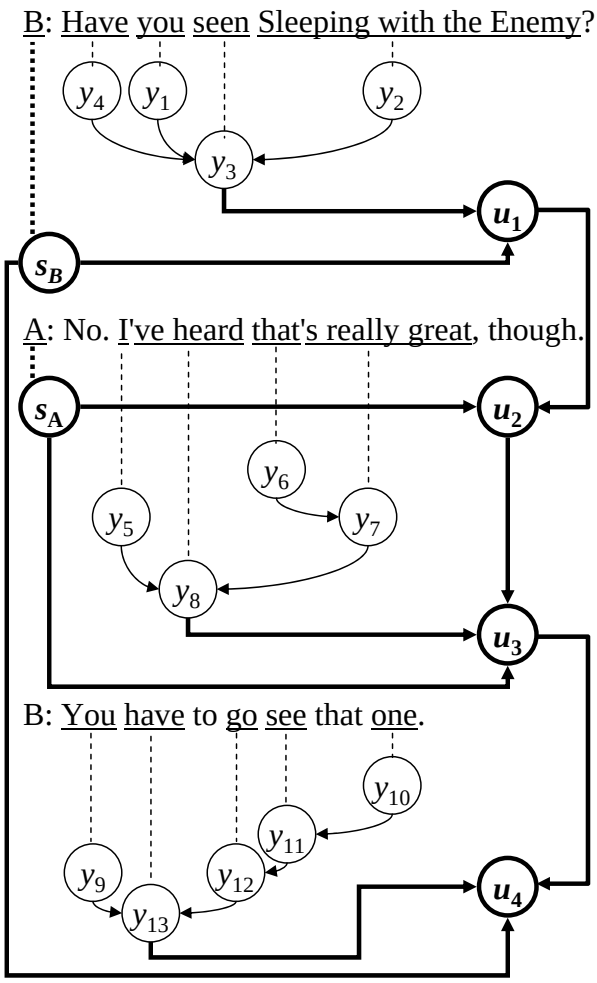
- **Speaker predicates**: we use a relational clause $speaker(\cdot, \cdot)$ to represent the speaker of each utterance, e.g., $speaker(u_1, s_B)$.

- **Content predicates**: we use a relational clause $content(\cdot, \cdot)$ to represent the content of each utterance, where the two arguments are the utterance term and the *head* term in the utterance, respectively. e.g., $content(u_3, y_8)$ (where $y_8 = heard$).

- **Utterance flow predicates**: we use a relational clause $follow(\cdot, \cdot)$ to connect each pair of adjacent utterances. e.g., $follow(u_2, u_1)$. We currently do not consider overlap in utterances, but our representation can be modified to handle this situation by introducing additional predicates.

Figure 5.5(b) shows the augmented representation for the same example in Figure 5.4, and Figure 5.5(a) shows the corresponding dependency structure together with conversation structure. The highlighted parts in these figures illustrate the newly introduced terms and predicates (relations). Because such representations of conversation segments take conversation structures into consideration, we call them *structural* representations. In contrast, we call previous representations without conversation structures (as in Figure 5.4) *basic* representations.

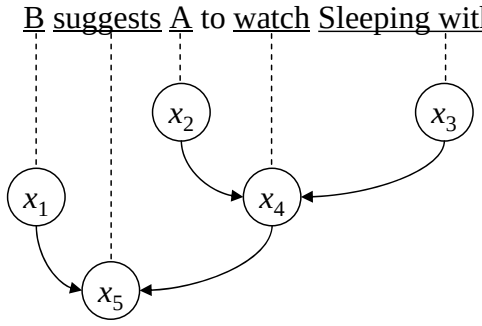
Our follow-up discussions are based on the same example in Figure 5.4(a) and



(a) dependency and conversation structures of the conversation segment

Terms	Clauses
s_A, s_B u_1=*yes-no-question*	***speaker***(u_1,s_B) ***content***(u_1, y_3)
y_1=*A*, y_2=*Sleeping with the Enemy* y_3=*seen*, y_4=*have*	*obj*(y_3,y_2) *subj*(y_3,y_1) *aux*(y_3,y_4)
u_2=*no-answer*	***speaker***(u_2,s_A) ***follow***(u_2,u_1)
u_3=*statement*	***speaker***(u_3,s_A) ***content***(u_3,y_8) ***follow***(u_3,u_2)
y_5=*A*, y_6=*that* y_7=*is really great* y_8=*have heard*	*subj*(y_7,y_6) *obj*(y_8,y_7) *subj*(y_8,y_5)
u_4=*opinion*	***speaker***(u_4,s_B) ***content***(u_4,y_{13}) ***follow***(u_4,u_3)
y_9=*A*, y_{10}=*one* y_{11}=*see*, y_{12}=*go* y_{13}=*have*	*obj*(y_{11},y_{10}) *obj*(y_{12},y_{11}) *obj*(y_{13},y_{12}) *subj*(y_{13},y_9)

(b) augmented representation of the conversation segment



(c) dependency structure of the hypothesis

Terms	Clauses
x_1=*B*, x_2=*A* x_3=*Sleeping with the Enemy* x_4=*watch* x_5=*suggests*	*subj*(x_4,x_2) *obj*(x_4,x_3) *subj*(x_5,x_1) *obj*(x_5,x_4)

(d) clause representation of the hypothesis

Figure 5.5: The conversation structure and augmented representation for the example in Figure 5.4

the representation in Figure 5.5. Therefore we also show the dependency structure and the corresponding clause representation for the hypothesis in Figure 5.5(c) and 5.5(d), respectively.

5.4.2 Modeling Conversation Structure in Alignment Model

Previously, our system is incapable of predicting entailments such as the one in Figure 5.4(a), because the hypothesis term *suggests* is not expressed explicitly in the premise, and thus the system cannot find an alignment in the premise for such a term. Instead, the conversation utterance of speaker B, *You have to go see that one*, constitutes the act of making a suggestion. Therefore, we propose to take conversation structure into consideration so as to solve this problem.

Specifically, with the structural representations of conversation segments, we incorporate those (pseudo) terms representing conversation utterances into our alignment model. We call the alignments involving such terms *pseudo alignments*. For example, Figure 5.4.2 gives a complete alignment between the premise terms and hypothesis terms in Figure 5.5, where $g(x_5, u_4) = 1$ is a pseudo alignment.

A pseudo alignment is identified between a hypothesis term x and a premise term u if they satisfy the following conditions:

1. x is a verb matching the dialogue act of u, e.g., $x_5 = suggests$ is a match of $u_4 = opinion$;

2. The subject of x matches the speaker of utterance u, e.g., the subject of x_5, $x_1 = B$, is a match of the speaker of u_4, which is s_B.

The match of subjects is pretty straightforward because the speaker of an utterance can only be either s_A or s_B. The match of verbs against dialogue acts is currently processed by a set of rules learned from the development data of conversation entailment. Each of such rules $V \sim U$ consists of two sets, a verb set V and a

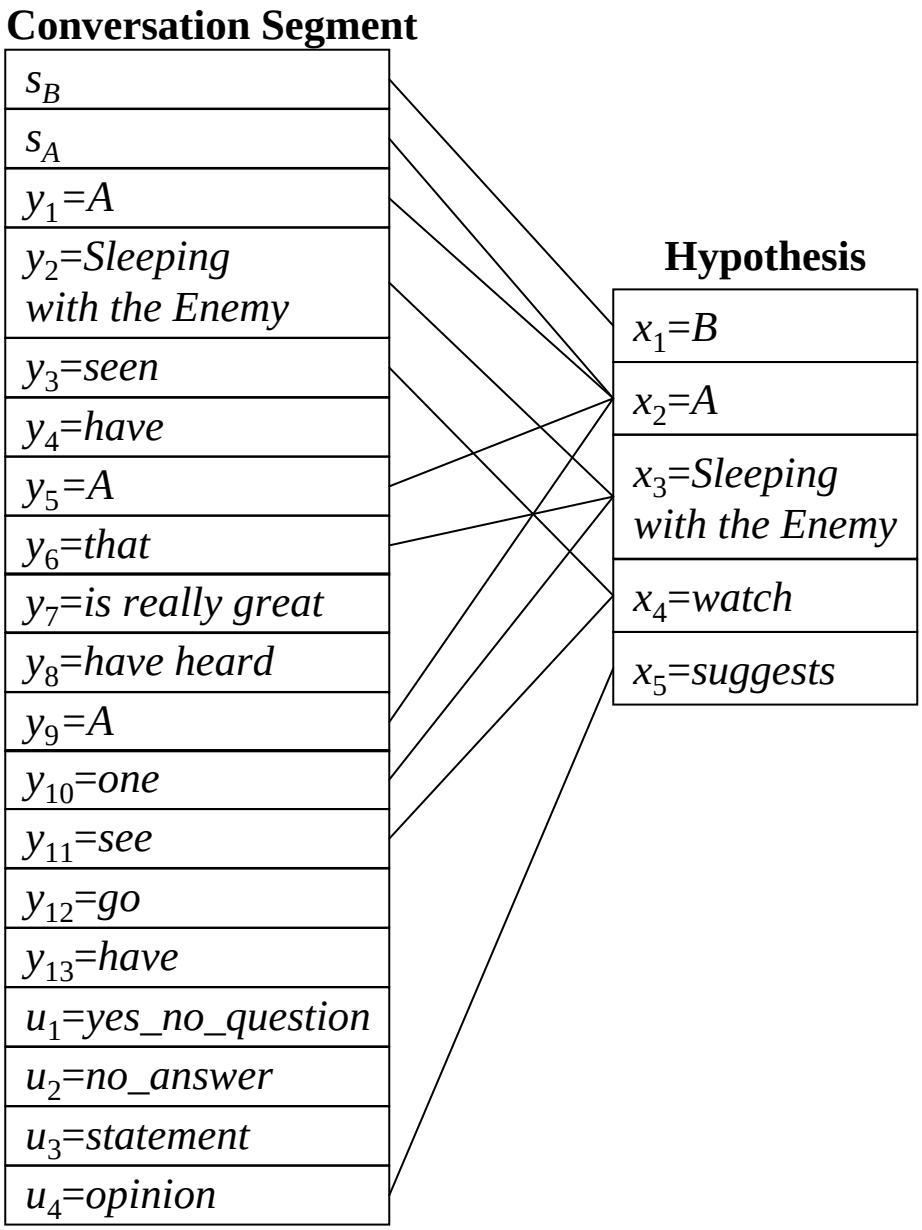


Figure 5.6: An alignment for the example in Figure 5.5

dialogue act set U, which means any verb in V can be a match to any dialogue act in U. Below are a few examples of such rules:

1. $\{think, believe, consider, find\} \sim \{statement, opinion\}$
2. $\{want, like\} \sim \{opinion, wh\text{-}question\}$
3. $\{agree\} \sim \{agree, acknowledge, appreciation\}$
4. $\{disagree\} \sim \{yes\text{-}no\text{-}question\}$

5.4.3 Evaluation

To investigate how the modeling of conversation structure helps our entailment system, in this section we evaluate the entailment system incorporating the structural features. The evaluation is again conducted on two tasks, the verb alignment task and the entailment prediction task.

Evaluation on Verb Alignment

Since we have introduced pseudo alignment in Section 5.4.2, now the ground truth for verb alignment is different from before when conversation structure was not incorporated in the representations. The current true alignments of verbs also include pseudo alignments, i.e., alignments between verbs terms in the hypotheses and (pseudo) utterances terms in the conversations. For this reason, the verb alignment for the system incorporating structural features of conversations can not be compared to that of a system without structural feature modeling. Therefore we evaluate the verb alignment of the system with structural feature modeling on its own.

Figure 5.7 shows the system's performance on verb alignment for both the development and the test data sets after modeling features of conversation structures. An overall trend is that when the threshold goes up, the system's performance does not

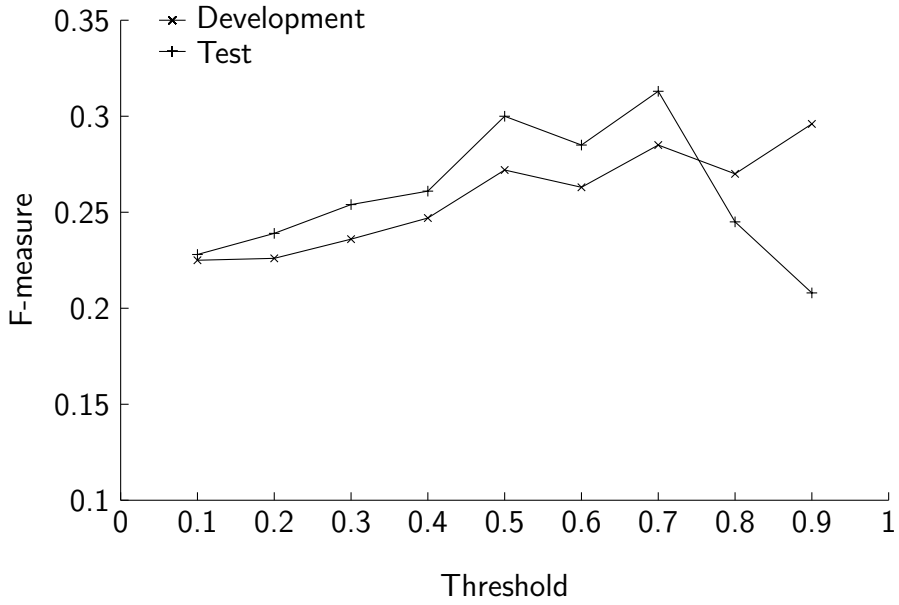


Figure 5.7: Evaluation of verb alignment for system modeling conversation structure features

decrease as much as the previous system without modeling conversation structures (see Figure 5.1), especially for the development data set. This is because our system takes a rule-based classification mechanism for pseudo alignments, so the recalls of pseudo alignments are not affected by high thresholds.

Figure 5.8 shows the alignment result broken down by different hypothesis types for both the development and the test data sets at threshold 0.7. A dramatic result we see in this figure is that the verb alignment performances for the *intent* hypotheses now exceed the performances for all other hypothesis types. This is what we expected – pseudo alignments help align the verb terms in *intent* hypotheses the most, since such hypotheses have many verbs (e.g., $x_5 = suggests$ in Figure 5.5(d)) that have to be aligned to pseudo terms of utterances with dialogue acts (e.g., $u_4 = opinion$ in Figure 5.5(b)).

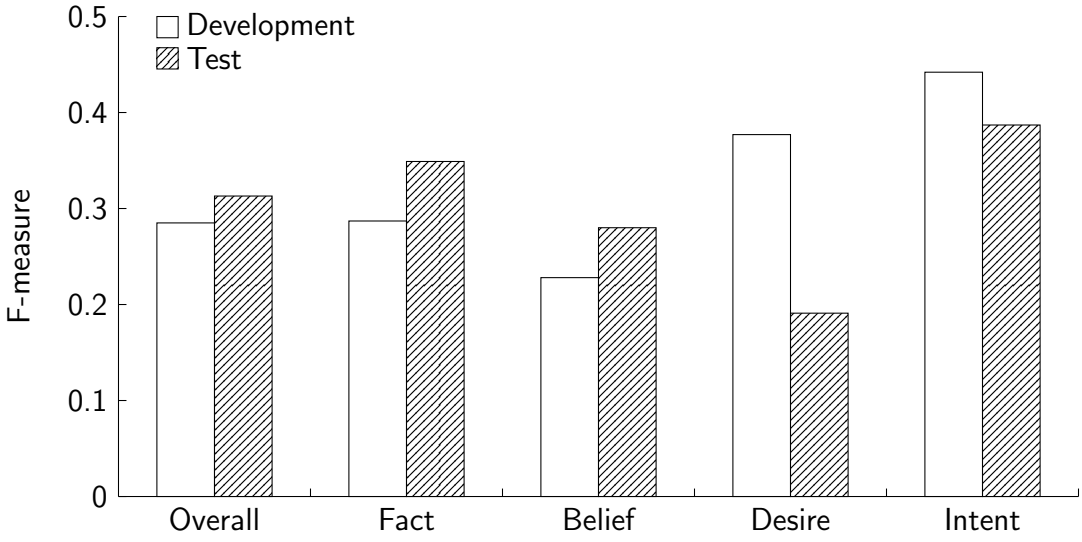


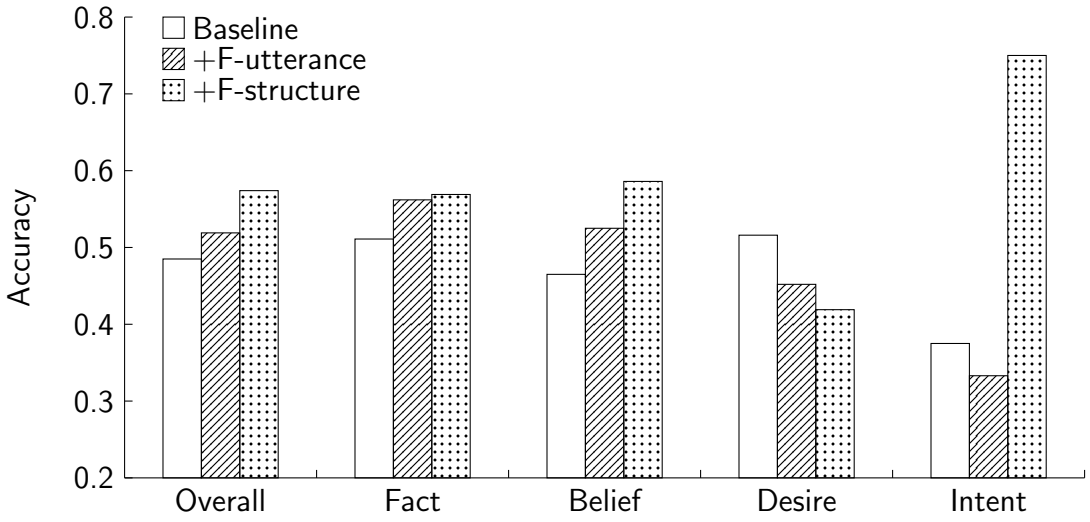
Figure 5.8: Evaluation of verb alignment by different hypothesis types for system modeling conversation structure features

Evaluation on Entailment Prediction

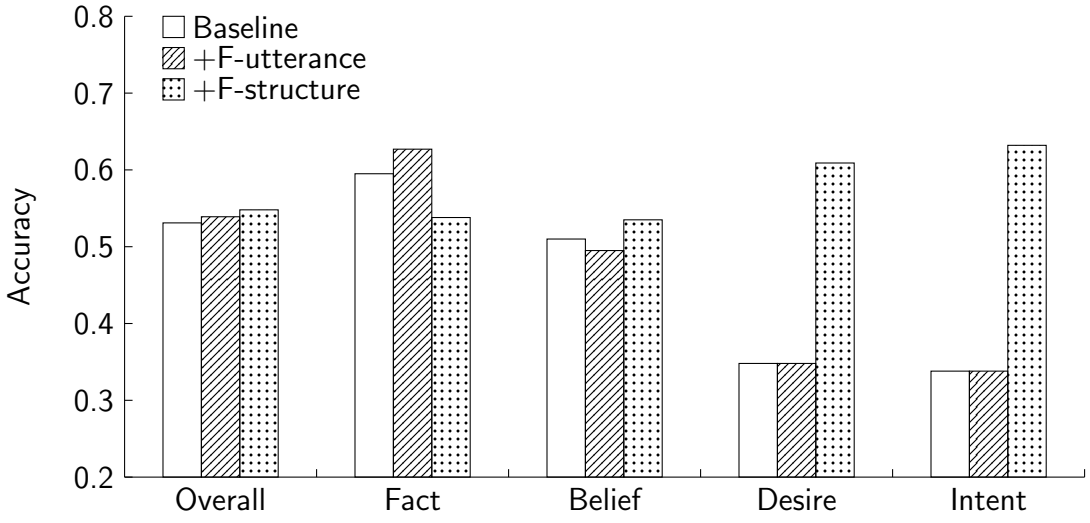
Figure 5.9(a) and 5.9(b) show the accuracies of entailment prediction for three systems on the development and test data sets. The three systems are a baseline system using models trained from textual entailment (*Baseline*), an improved system modeling linguistic features in conversation utterances (*+F-utterance*), and a further improved system incorporating features of conversation structures (*+F-structure*).

Overall speaking, the system modeling conversation structure has limited improvement compared to other two systems. The improvement is more noticeable on the development data, since the model to capture pseudo alignments is learned from the same data set.

However, if we break down the same evaluation results by different types of hypotheses (which is also shown in Figure 5.9), we can see that the system modeling conversation structure features increases the prediction accuracy significantly for *intent* type of hypotheses (McNemar's test, $p < 0.05$). This is consistent with what we found in evaluating verb alignments, i.e., the incorporation of pseudo alignments is



(a) On development data



(b) On test data

Figure 5.9: Evaluation of entailment prediction for system modeling conversation structure features

most effective for hypotheses of *intent*s.

It should also be noted that after incorporating features modeling conversation structures, the whole system is re-trained to maximize the performance for all hypothesis types. As a trade-off, the performance on some subset of examples is sacrificed (decreased). For example, in Figure 5.9(b), for the *fact* type the performance of the *+F-structure* system is decreased on the test data compared to the *+F-utterance* system. For the *desire* type the performance is increased on the test data but decreased on the development data.

So why in the end does the performance on some examples decrease? We further investigated the changes brought into the system by modeling conversation structures. We found that structural modeling creates more connectivity for language constituents that were not connected before. For example:

Premise:

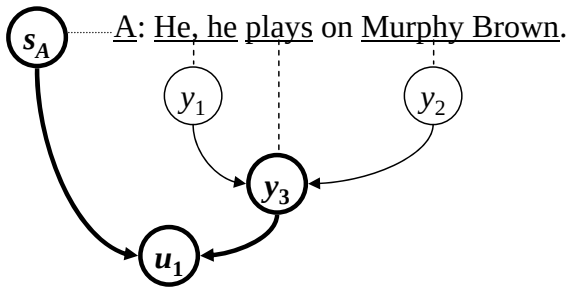
A: He, he plays on *Murphy Brown*.

Hypothesis:

A plays on *Murphy Brown*.

The speaker A is not in the utterance of conversation, so our previous system would not find an alignment for the hypothesis term A, and thus predicts the entailment to be false, which is a correct prediction. However, after introducing the modeling of conversation structure, we identify *plays* as the content of the utterance in the conversation segment, and A as the speaker of that utterance, as we show in Figure 5.10. Thus a link between *plays* and A is established, i.e., $y_3 \sim u_1 \sim s_A$ in Figure 5.10(a).

In this case, a correct prediction of the false entailment has to recognize in Figure 5.10(a) that the relationship between y_3 and s_A is not a verb-subject relation.



(a) The dependency and conversation structures of the conversation segment

Terms	Clauses
u_1=*statement* s_A=*A*	*speaker*(u_1,s_A) *content*(u_1,y_3)
y_1=*he* y_2=*Murphy Brown* y_3=*plays*	*obj*(y_3,y_2) *subj*(y_3,y_1)

(b) The structural representation of the conversation segment

Figure 5.10: An example of measuring the relationship between two terms by their distance (the highlighted distance between y_3 and s_A is 2)

However, our primitive entailment models only recognize the distance between language constituents as their semantic relationship, i.e., for alignment model in Section 3.3.1, we use distance to model the relations between a verb and its arguments (subject or object), and for inference model in Section 3.3.2, we use distance to model any relation between two terms. As a result, as the distance between y_3 and A in Figure 5.10(a) is 2, the alignment model would recognize A as an argument of y_3, and the inference model would use it to infer the hypothesis clause $subject(plays, A)$.

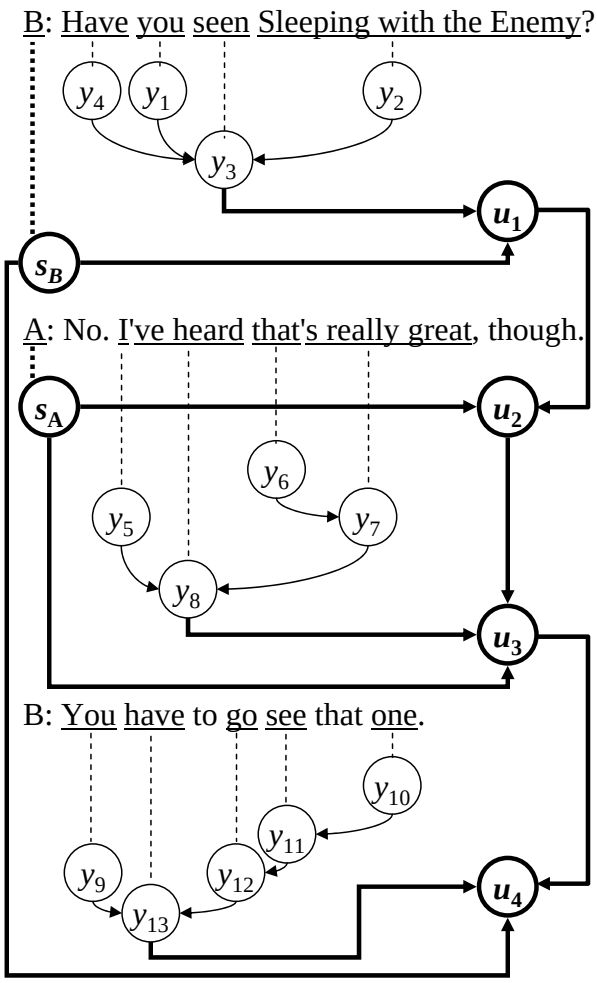
Therefore, it is critical to develop a better approach of modeling semantic relations between language constituents, to improve our models for both alignment classification and inference recognition.

Chapter 6

Enhanced Models for Conversation Entailment

In Section 5.4.3 we have pointed out that the current models in our entailment system are very simple. A major inadequacy in these models is they simply use the distance between two language constituents to model the semantic relationship between them. More specifically, in the alignment model, we use distance to model the relationship between a verb and its arguments (subject or object). And in the inference model, we use distance to model the relationship between any two terms.

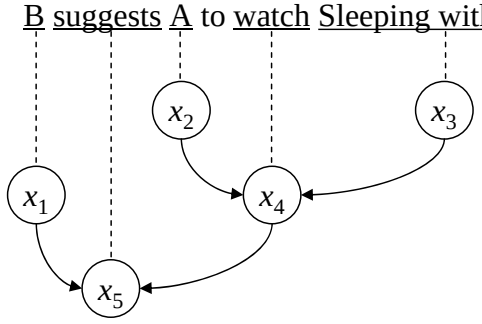
To address this problem, in this chapter we aim at enhancing the entailment models by incorporating more semantics into the modeling of long distance relationship between language constituents [93]. We first describe two approaches of modeling long distance relationship, and then discuss about how these approaches are employed in our entailment models. For the convenience of discussion, we copy Figure 5.5 as Figure 6.1.



(a) dependency and conversation structures of the conversation segment

Terms	Clauses
s_A, s_B u_1=*yes-no-question*	***speaker***(u_1,s_B) ***content***(u_1, y_3)
y_1=*A*, y_2=*Sleeping with the Enemy* y_3=*seen*, y_4=*have*	*obj*(y_3,y_2) *subj*(y_3,y_1) *aux*(y_3,y_4)
u_2=*no-answer*	***speaker***(u_2,s_A) ***follow***(u_2,u_1)
u_3=*statement*	***speaker***(u_3,s_A) ***content***(u_3,y_8) ***follow***(u_3,u_2)
y_5=*A*, y_6=*that* y_7=*is really great* y_8=*have heard*	*subj*(y_7,y_6) *obj*(y_8,y_7) *subj*(y_8,y_5)
u_4=*opinion*	***speaker***(u_4,s_B) ***content***(u_4,y_{13}) ***follow***(u_4,u_3)
y_9=*A*, y_{10}=*one* y_{11}=*see*, y_{12}=*go* y_{13}=*have*	*obj*(y_{11},y_{10}) *obj*(y_{12},y_{11}) *obj*(y_{13},y_{12}) *subj*(y_{13},y_9)

(b) augmented representation of the conversation segment



(c) dependency structure of the hypothesis

Terms	Clauses
x_1=*B*, x_2=*A* x_3=*Sleeping with the Enemy* x_4=*watch* x_5=*suggests*	*subj*(x_4,x_2) *obj*(x_4,x_3) *subj*(x_5,x_1) *obj*(x_5,x_4)

(d) clause representation of the hypothesis

Figure 6.1: A copy of Figure 5.5: the structural representation of a conversation segment and the corresponding hypothesis

6.1 Modeling Long Distance Relationship

Relationship can exist between any two language constituents in a discourse, even when there is not a direct syntactical relation between them. For example, in Figure 6.1(a), the term $y_9 = you$ is not the syntactic subject of the term $y_{11} = see$ (i.e., we do not have $subject(y_{11}, y_9)$). However, if we try to identify the arguments for the verb $y_{11} = see$, we can see that $y_9 = you$ is its logical subject. We call such a relation between two terms a *long distance relation (LDR).*

This raises a question of how we can find the logical relation between the two terms (e.g., $logic\_subject(y_{11}, y_9)$), given the current representation of the conversation segment (i.e, dependency plus conversation structures).

6.1.1 Implicit Modeling of Long Distance Relationship

Our previous approach is to use the distance between the two terms in the structural representation as the modeling of their long distance relationship. For example, in Figure 6.1(a), the distance between y_{11} and y_9 is 3. We call this approach the *implicit modeling* of long distance relationship.

The rationale behind the implicit modeling approach is, the closer that two terms are in the dependency and conversation structures, the more likely there is a relationship between them. And, as a basic approach, we do not distinguish what type that relationship is. For example, the hypothesis terms x_4 and x_2 in Figure 6.1(d) has a relationship of $subject(x_4, x_2)$. This relationship will be determined as entailed from y_{11} and y_9 in the premise, if x_4 and x_2 are aligned to y_{11} and y_9 respectively, and the distance between y_{11} and y_9 is close. This decision is made regardless of whether the relationship between y_{11} and y_9 is $subject(y_{11}, y_9)$.

The advantage of the implicit modeling approach is that it is easy to implement based on the dependency and conversation structures. However, its limitation is

that the distance measure does not capture the semantics of relation types between language constituents. For example, in Figure 5.10, the distance between the terms y_3 and s_A is 2, so the algorithm identified there is a relationship between them. However, as the type of this relationship is not identified, our entailment system would mistakenly use it to infer relations like $subject(\cdot, \cdot)$.

6.1.2 Explicit Modeling of Long Distance Relationship

We noticed that the identification of relation types such as $subject(\cdot, \cdot)$ is very much like identifying the arguments of a verb, e.g., whether an entity is the subject of a verb. In similar language processing tasks such as semantic role labeling [74], previous work has often used the path from one constituent to the other in a syntactic parse tree as a feature to identify the verb-argument relationship. Hence we adopt the same idea here, to use the path between two terms in the dependency structure (augmented with conversation structure) to model the long distance relationship.

Specifically, a path from one term to another in a dependency/conversation structure is defined as a series of labels representing the vertices and edges connecting them:

$$v_1 e_1 \dots v_{l-1} e_{l-1} v_l$$

where $v_1, \dots, v_l$ are the labels of vertices on the path, and $e_1, \dots, e_{l-1}$ are the labels of edges on the path. In our experiment we label the vertices by one of the three types: noun (N), verb (V), or utterance (U); and label the edges by their directions: forward ($\rightarrow$) or backward ($\leftarrow$). For example, in Figure 6.1(a), the path from y_{11} to y_9 is

$$V \rightarrow V \rightarrow V \leftarrow N$$

and in Figure 5.10(a) the path from y_3 to s_A is

$$V \rightarrow U \leftarrow N$$

Although various labels can be designed to describe the vertices and edges on a path, our criteria of choosing such labeling system are that

1. They are adequately detailed to capture the semantics of different types of relations. For example, it should be differentiated that $V \rightarrow V \rightarrow V \leftarrow N$ models a verb-subject relationship, while $V \rightarrow U \leftarrow N$ does not.

2. They are also abstracted to certain extent (i.e., not overly detailed), in order for the modeling to be generalizable. For example, if we describe the path from x_{11} to x_9 in Figure 6.1(a) as $see \xrightarrow{object} go \xrightarrow{object} have \xleftarrow{subject} you$, this pattern may not be seen again in other examples.

6.2 Modeling Long Distance Relationship in the Alignment Model

In Chapter 3 we have described the mechanism of how the alignment model works in our entailment system. Specifically, in Section 3.3.1 we have described the feature sets used to train the alignment models for nouns and verbs. A verb alignment model classifies for two verbs, x from a hypothesis and y from a premise, whether they are aligned or not. Two important features in the verb alignment model are whether the arguments (i.e., subjects and objects) of x and y are consistent.

In this section we first give a brief review of how the argument consistencies are modeled in our previous system, then propose an enhanced model of the argument consistencies, and finally evaluate both modeling methods and compare their perfor-

mances.

6.2.1 Implicit Modeling of Long Distance Relationship in the Verb Alignment Model

The previous approach models the argument consistency of two verbs based on implicit modeling of the relationship between a verb and its aligned subject/object.

Specifically, given a pair of verb terms (x, y) where x is from the hypothesis and y is from the premise, let s_x be the subject of x in the hypothesis, and let s_y be the aligned entity of s_x in the premise (in case of multiple alignments, s_y is the one closest to y). The subject consistency of the verbs (x, y) is then modeled by the long distance relationship between s_y and y. For implicit modeling of LDR, such relationship is measured by the distance between s_y and y in the (augmented) dependency structure of the premise.

For example, in Figure 6.1, to decide whether the hypothesis term $x_4 = watch$ and the premise term $y_{11} = see$ should be aligned, we first identify the subject of x_4 in the hypothesis, which is $x_2 = A$. We then look for x_2's alignments in the premise, among which $y_9 = you$ is the closest to x_{11}. In Figure 6.1(a), we find the distance between y_{11} and y_9 is 3.

Similarly, the distance between a verb and its aligned object is used as a measure of the object consistency.

By implicit modeling of the long distance relationship in the alignment model, the feature values of argument consistencies are quantitative. And since all other features used in the alignment model (see Section 3.3.1) are either quantitative or binary, we trained a discriminative binary classification model (e.g. logistic regression model) to classify verb alignments.

The limitation of such an alignment model is that the implicit modeling of LDR

does not capture the semantic relationship between a verb and its aligned subject or object. For example, as we discussed in Section 5.4.3, the implicit alignment model would also identify the term s_A in Figure 5.10 as the subject of y_3.

6.2.2 Explicit Modeling of Long Distance Relationship in the Verb Alignment Model

In order to model more semantics in the relationship between a verb and its aligned subject/object, we adopt the explicit modeling of long distance relationship.

Given a pair of verb terms (x, y), let s_x be the subject of x and s_y be the aligned entity of s_x in the premise closest to y, we use explicit modeling of the long distance relationship between y and s_y as the feature to capture subject consistency. That is, we use a string to describe the path from y to s_y. The string description of a path is defined in Section 6.1.2. For example, in Figure 6.1(a), the path from y_{11} to y_9 is $V \rightarrow V \rightarrow V \leftarrow N$.

Such explicit modeling is used to capture both the subject consistency and the object consistency. Since the string representation of paths is not quantitative, we cannot plot the data instances with this feature into a measurable feature space. In order to use a discriminative model such as logistic regression to do the classification, traditional way of quantifying a string feature is to convert it to p binary features, where p is the possible number of values of the original string feature. However, in our case the number of values for the path feature can be very large, in comparison the size of our data set is relatively small. As a result, this will cause a severe sparse data problem.

Therefore, we use an instance-based classification model (e.g., k-nearest neighbour) instead of the discriminative model. An instance-based model requires only a distance measure between any two particular instances. Suppose that for any fea-

ture f_i, we have a distance measure between any two of its values, $v(f_i)$ and $w(f_i)$, then for any two instances a and b with n features $f_1 \dots f_n$, where the feature values for instance a are $v_1(f_1) \dots v_n(f_n)$ and the feature values for instance b are $w_1(f_1) \dots w_n(f_n)$, we can calculate the distance between a and b by their Euclidean distance:

$$dist(a, b) = \sqrt{\frac{\sum_{i=1}^{n} dist\left(v_i(f_i), w_i(f_i)\right)^2}{n}}$$

For binary features such as *verb be identification*, *string equality*, and *stemmed equality* in Section 3.3.1, the distance between two values is whether the two values are the same:

$$dist\left(v_i(f_i), w_i(f_i)\right) = \begin{cases} 1 & \text{if } v_i(f_i) = w_i(f_i) \\ 0 & \text{otherwise} \end{cases}$$

For quantitative features such as *WordNet similarity* and *distributional similarity* in Section 3.3.1, the distance between two values is the absolute value of their difference:

$$dist\left(v_i(f_i), w_i(f_i)\right) = |v_i(f_i) - w_i(f_i)|$$

And for string features such as the subject/object consistency (with explicit modeling of LDR), the distance between two values is their minimal string edit distance (Levenshtein distance).

6.2.3 Evaluation of LDR Modelings in Alignment Models

We evaluate two alignment models with different modeling of long distance relationship, one with implicit modeling and one with explicit modeling. The implicit alignment model is the same as we evaluated in Sections 4.4, 5.2.4, and 5.4.3, and the explicit alignment model is trained using a k-nearest neighbour model (described in Section 6.2.2) from the development data set. So we compare their performances of

verb alignment only on the test data of conversation entailment.

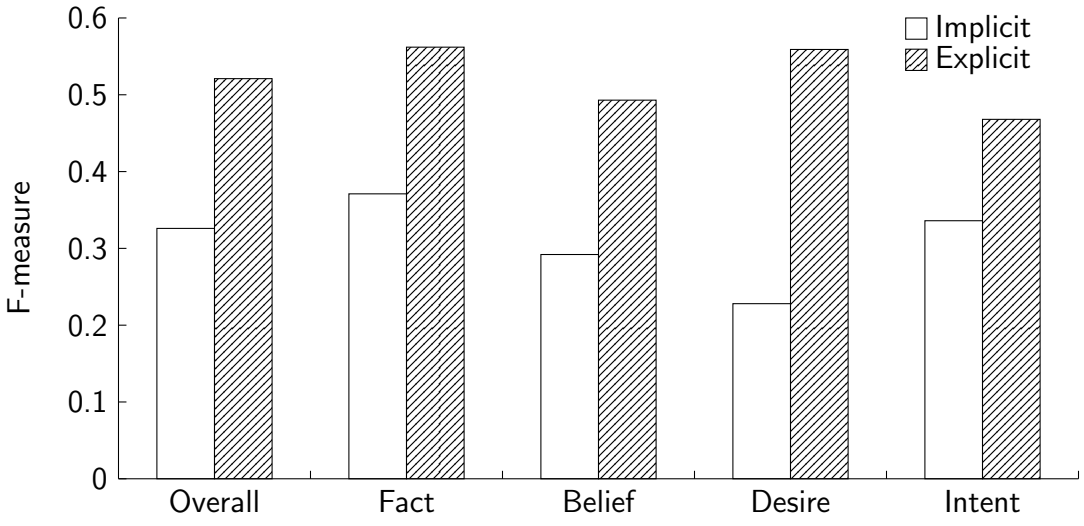
Figure 6.2(a) compares two alignment models based on basic representation of conversation segments, and Figure 6.2(b) compares two alignment models based on structural representation of conversation segments. The performances of the implicit alignment model in these two figures are the same as *+F-utterance* in Figure 5.2(b) and *Test* in Figure 5.8, respectively. Also similar to the discussion in Section 5.4.3, the results between Figure 6.2(a) and Figure 6.2(b) are not meant to be compared directly, since they involve different numbers of alignment instances.

Overall speaking, the explicit model outperforms the implicit model. This suggests that the explicit modeling of long distance relationship between verbs and their arguments works better than the implicit modeling used in the previous alignment model. Furthermore, as we break down the results by different types of hypotheses, we can see that the improvements are more noticeable when hypothesis types are *fact*, *belief*, and *desire*, while the improvement is the least for *intent* hypotheses. This is because that a large portion of verbs in *intent* hypotheses are aligned to pseudo utterance terms in the premise, which are handled by the rule-based pseudo alignment model rather than the verb alignment model described and enhanced in this section.

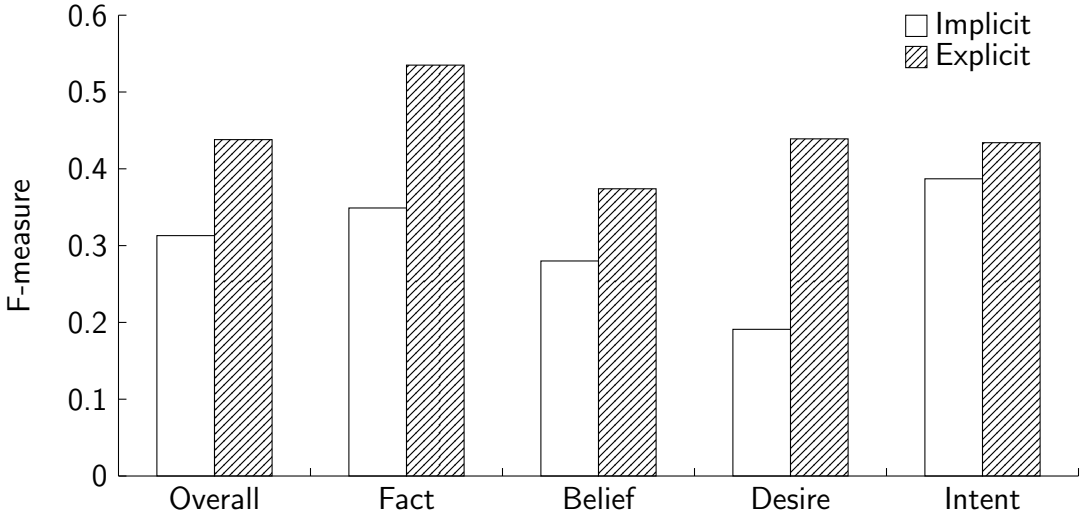
6.3 Modeling Long Distance Relationship in the Inference Model

In Section 3.1.3 we have formulated the inference model as to predict the probability that a clause s_j in the hypothesis is entailed from a set of clauses $d_1 \dots d_m$ in the premise, given an alignment scheme g between the terms in the hypothesis and the terms in the premise:

$$P(d_1 d_2 \dots d_m \vDash s_j | d_1, d_2, \dots, d_m, s_j, g)$$



(a) Based on basic representation



(b) Based on structural representation

Figure 6.2: Evaluation of verb alignment with different modelings of long distance relationship

In Section 3.3.2 we have described the feature sets that we used to train our inference models, which are distinguished between the inference of property clauses and the inference of relational clauses. The inference of relational clauses involves the modeling of long distance relationship.

In this section we first give a review on the implicit modeling of long distance relationship that is previously used in the relational inference model, and then discuss about how the model can be enhanced by the explicit modeling of LDR. After that, the enhanced model of relational inference is evaluated and compared to the original inference model.

6.3.1 Implicit Modeling of Long Distance Relationship in the Relational Inference Model

If the hypothesis clause s_j is a relational clause, that means it takes two arguments (hypothesis terms). We denote it as $s_j(x_1, x_2)$. To predict whether it is entailed from the premise, we first find the counterparts (aligned terms) of x_1 and x_2 in the premise:

$$D'_1 = \{y | y \in D, g(x_1, y) = 1\}$$

$$D'_2 = \{y | y \in D, g(x_2, y) = 1\}$$

and then get the closest pair of terms (y_1^*, y_2^*) from these two sets, i.e., the distance between y_1^* and y_2^* in the (augmented) dependency structure of premise is the smallest among any $y_1 \in D'_1$ and $y_2 \in D'_2$.

For example, in Figure 6.1, if we want to infer whether the hypothesis clause $object(x_5, x_4)$ is entailed, we find the alignments for $x_5 = suggests$ and $x_4 = watch$, which are $\{u_4 = opinion\}$ and $\{y_3 = seen, y_{11} = see\}$ respectively. In Figure 6.1(a), the distance between u_4 and y_3 is 4, and the distance between u_4 and y_{11} is 3, so the

closet pair of terms between these two sets is u_4 and y_{11}.

So the inference decision on s_j should be determined by the long distance relationship between the premise terms y_1^* and y_2^*, i.e., whether (1) there is a relationship between y_1^* and y_2^*; and (2) whether such relationship is the same as s_j, which describes the relationship between hypothesis terms x_1 and x_2.

Using implicit modeling of long distance relationship, we predict whether s_j is inferred only by the distance between y_1^* and y_2^*. The smaller this distance is, the more likely these two terms have a direct relationship. Though such an assumption is reasonable, and the implicit modeling addresses to certain extent the first question above, however, it does not address the second question: whether the relationship between y_1^* and y_2^* is the same as described by s_j.

6.3.2 Explicit Modeling of Long Distance Relationship in the Relational Inference Model

In order to identify the relationship between y_1^* and y_2^*, we need to capture more semantics in the relationship between the two terms. As an enhanced model, we use explicit modeling instead of the implicit one to model the long distance relationship between y_1^* and y_2^*.

In Figure 6.1(a), for example, the explicit modeling of long distance relationship between u_4 and y_{11} is

$$U \leftarrow V \leftarrow V \leftarrow V$$

Similar to Section 6.2.2, we use an instance-based model (e.g. k-nearest neighbour) to classify the inference decision on $s_j(x_1, x_2)$. The explicit modeling of long distance relationship between y_1^* and y_2^* is used as a feature in the classification model. Such a feature has values in string forms, and the distance between two of its values can be calculated by their minimal string edit distance (as discussed in Section 6.2.2).

Additionally, the instance-based classification model enables us to add an additional set of nominal features into the classifier. Below is the list of additional features used in our system (given that the hypothesis clause to be inferred is $s_j(x_1, x_2)$ and the closest pair of terms aligned to x_1 and x_2 in the premise are y_1^* and y_2^* respectively):

1. The types (noun/verb/utterance) of x_1, x_2, y_1^*, and y_2^*;
2. The type of relation between x_1 and x_2, for example, *object* in $object(x_1, x_2)$;
3. The order (i.e., *before* or *after*) between x_1 and x_2, and between y_1^* and y_2^*;
4. The specific type of the hypothesis (*fact*/*belief*/*desire*/*intent*).

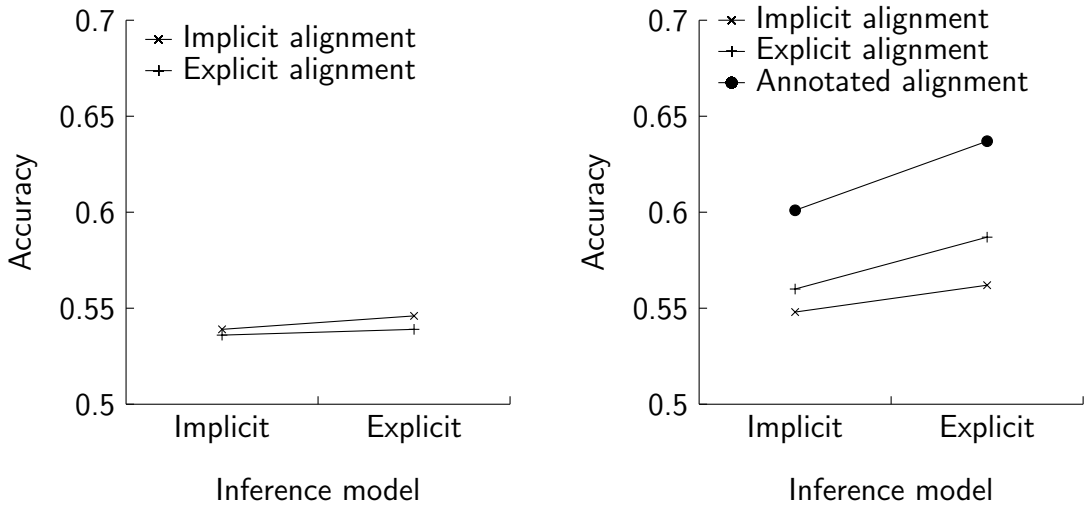
The distance between two values $v_i(f_i)$ and $w_i(f_i)$ of a nominal feature is estimated based on whether the two values are the same (similar to binary features in Section 6.2.2):

$$dist\left(v_i(f_i), w_i(f_i)\right) = \begin{cases} 1 & \text{if } v_i(f_i) = w_i(f_i) \\ 0 & \text{otherwise} \end{cases}$$

6.3.3 Evaluation of LDR Modelings in Inference Models

In this section we compare the performances of two inference models, one using implicit modeling of long distance relationship and one using explicit modeling. The explicit inference model is trained from the development data of conversation entailment using features described in Section 6.3.2. So the evaluation and comparison are conducted only on the test data set.

Figure 6.3(a) shows the prediction results (accuracies) of the two inference models based on the basic representation of conversation segments, and Figure 6.3(b) shows the results based on the structural representation of conversation segments. In both figures we show inference results with different configurations of the alignment model



(a) Based on basic representation

(b) Based on structural representation

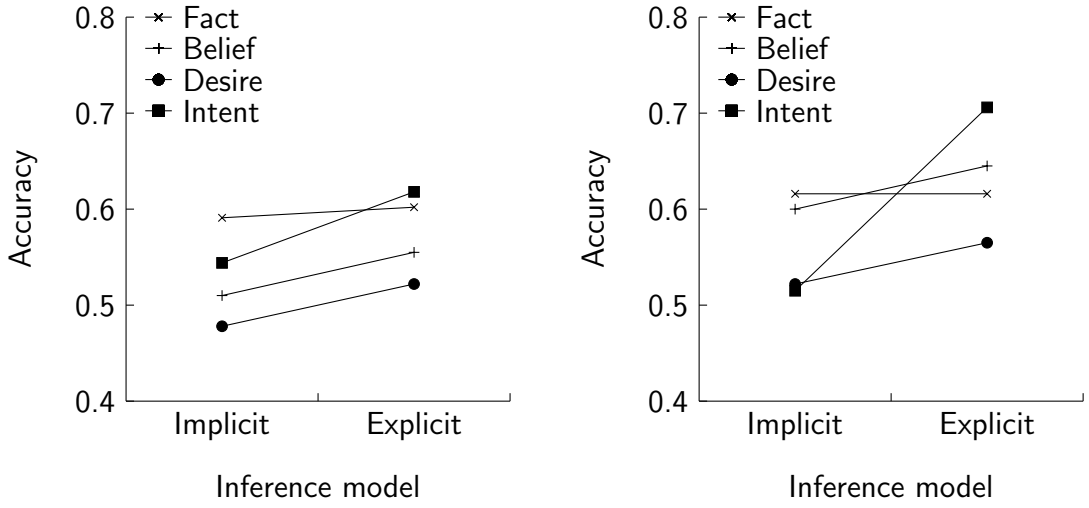
Figure 6.3: Evaluation of inference models with different modelings of long distance relationship

(implicit or explicit). Additionally, in Figure 6.3(b) we also show the results based on manual annotations of alignments.

In Figure 6.3(a) when we use the basic representation of conversation segments, the two inference models perform almost the same. This illustrates that when conversation structural is missing from the representation, the explicit modeling of LDR in the inference model offers no significant advantage compared to implicit modeling.

But when conversation structures are incorporated in the representation of conversation segments, as we show in Figure 6.3(b), the explicit inference model consistently performs better than the implicit model. The difference is statistically significant when the alignment model is also explicit or annotated (McNemar's test, $p < 0.05$).

The best performance of our system on test data, without using manual annotations of alignments, is achieved under the configuration of the structural representation of conversation segments, the explicit alignment model, and the explicit inference model. The accuracy is 58.7%. Compared to the natural baseline which always predicts the majority class (53.1% accuracy on our testing data), our system achieves a



(a) Based on structural representation and explicit alignment model

(b) Based on structural representation and annotated alignments

Figure 6.4: Evaluation of inference models with different LDR modelings for different hypothesis types

significantly better result (z-test, $p < 0.05$).

We further break down the evaluation results of two settings in Figure 6.3(b) (one with the structural representation and the explicit alignment model and one with the structural representation and the annotated alignments) by different types of hypotheses. The results are shown in Figure 6.4. We can see that the explicit inference model performs better than the implicit inference model in almost every subcategory. For both settings in Figure 6.4(a) and Figure 6.4(b), the improvements by explicit modeling of LDR are most prominent for the *intent* type of hypotheses.

It is interesting to see the difference in the system performances on different types of hypotheses, for example, *fact* and *intent*. In Section 6.2.3 we discovered that the *fact* hypotheses benefits more from explicit LDR modeling in the alignment model than the *intent* hypotheses. While we evaluate different LDR modelings in the inference model in this section, the findings are the opposite. That means for *fact* hypotheses, the more benefit from incorporating explicit modeling of long distance relationship appears at the alignment stage, while for *intent* hypotheses, the bene-

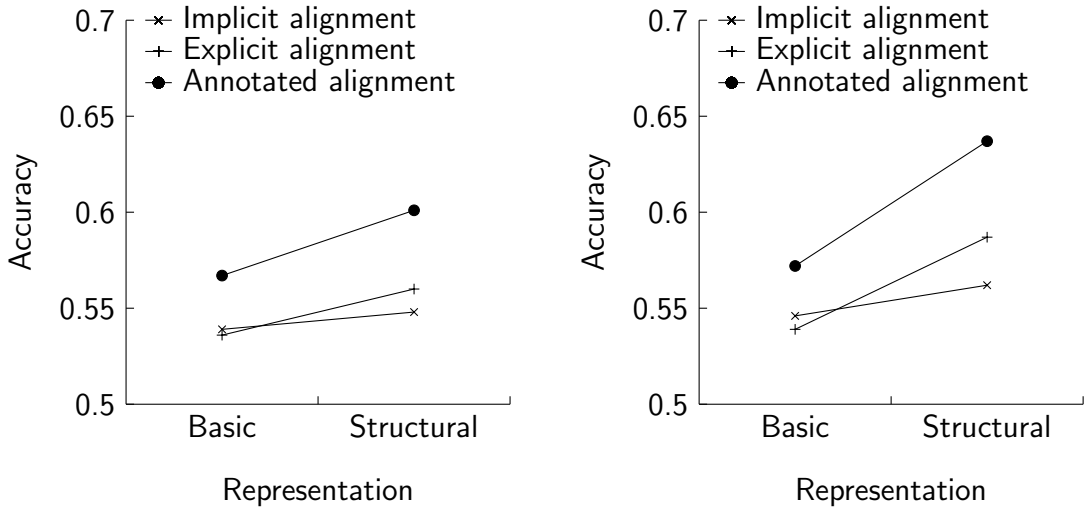
fit of explicitly modeling long distance relationship mostly happens at the inference stage.

This observation shows that the effects of different types of modeling may vary for different types of hypotheses, which indicates that hypothesis type dependent models may be beneficial. However, since the current amount of training data is relatively small, our initial investigation has not yielded significant improvement. Nonetheless, this still remains a promising direction when large training data becomes available.

6.4 Interaction of Entailment Components

In Section 6.2.3 we have evaluated different implementations of alignment models and in Section 6.3.3 we have evaluated different implementations of inference models. We conducted both evaluations under different settings of conversation representations: a basic representation of conversation utterances only, and an augmented representation incorporating conversation structures. We have noticed that there is an interaction between these different components of our entailment system. For example, in Section 6.3.3, we found that the effect of explicit modeling of long distance relationship in the inference model is dependent on the incorporation of conversation structure in the clause representation.

In this section we further study the interaction between different entailment components, including different representations of conversation segments and different modelings of long distance relationship in alignment and inference models. Specifically, we want to study how the change of one component might influence the entailment results, under various configurations of the other components.



(a) With implicit inference model

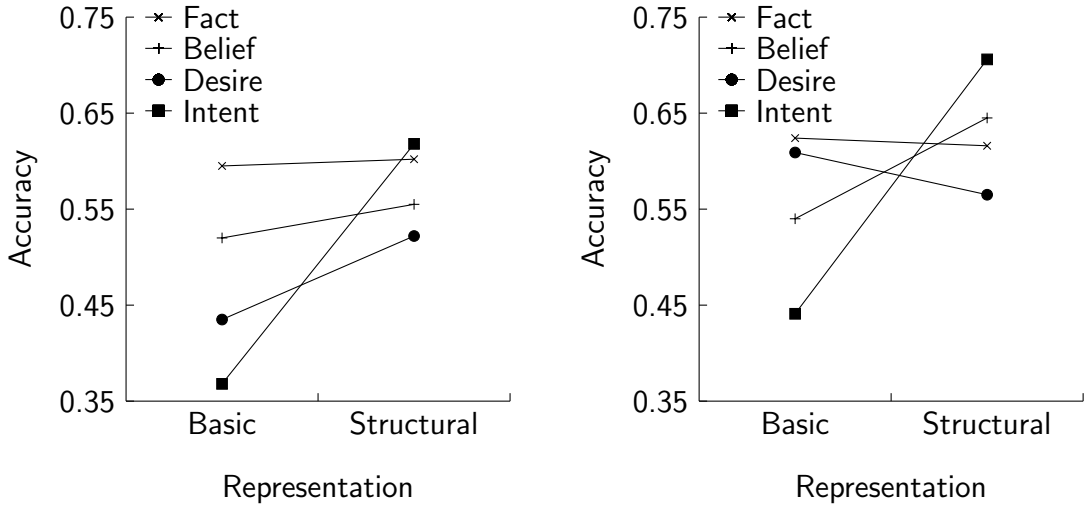
(b) With explicit inference model

Figure 6.5: Effect of different representations of conversation segments on entailment performance

6.4.1 The Effect of Conversation Representations

In Section 5.4.3 we have evaluated the effect of structural representation on entailment prediction, with implicit modeling of long distance relationship in both alignment and inference models. In this section we study how the system's performance is affected by representations of conversation segments, with variations of different long-distance-relationship modeling in alignment and inference models. Specifically, we want to compare the basic and structural representations under implicit and explicit modelings of long distance relationship in the alignment model and in the inference model.

Figure 6.5(a) shows the comparison of entailment results while using two different conversation representations, under the setting of implicit LDR modeling in inference model. Figure 6.5(b) shows the same comparison under the setting of explicit LDR modeling in inference model. In each of these figures, we conduct the comparison under three settings of alignment models: one with implicit modeling of LDR, one with explicit model of LDR, and one with annotated alignments. We can see that



(a) With explicit alignment model and explicit inference model

(b) With annotated alignments and explicit inference model

Figure 6.6: Effect of different conversation representations for different hypothesis types

for all six different configurations of alignment and inference models, the structural representation consistently yields better entailment performance than the basic representation.

In addition, for two of the settings in Figure 6.5(b), namely, explicit/explicit and annotated/explicit for alignment/inference models, the improvement brought by structural representation compared to basic representation is statistically significant (McNemar's test, $p < 0.01$). Considering the fact that these two configurations demonstrate bigger advantage of structural representation than other configurations, we may conclude that the structural representation has the most prominent advantage over the basic representation when it is used together with downstream components (alignment and inference models) that take into consideration of shallow semantics (i.e., explicit modeling of long distance relationship).

We further break down the comparison results under these two configurations (explicit/explicit and annotated/explicit) by different types of hypotheses, as shown in Figure 6.6. In both Figure 6.6(a) and Figure 6.6(b), the performance difference

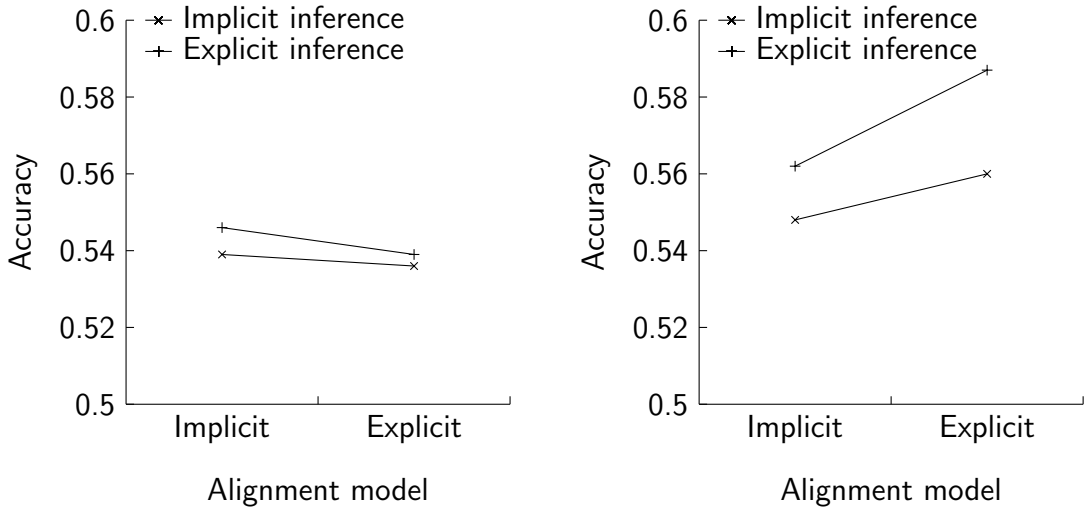
between the basic representation and the structural representation is not significant for hypotheses of *fact*, *belief*, and *desire*. However, for hypotheses of *intent*, the structural representation shows significant advantage over the basic representation (McNemar's test, $p < 0.001$).

This is consistent with what we found in Section 5.4.3: no matter what entailment models are used (implicit, explicit, or annotated), the improvement brought by structural representation mainly comes from *intent* type of hypotheses. Such an observation is not surprising, since most hypotheses in other subcategories (especially *fact* ones) can be inferred directly from conversation utterances.

6.4.2 The Effect of Alignment Models

Different from the study in Section 6.2.3, where alignment models with different modelings of long distance relationship are evaluated by the alignment results they produce, in this section we study how the system's entailment performance is affected by using different alignment models. Specifically, we want to compare the implicit and explicit alignment models under various settings of conversation representations and inference models.

Figure 6.7(a) compares the entailment results while using different alignment models, based on the basic representation of conversation segments. Figure 6.7(b) shows the same comparison based on the structural representation of conversation segments. Different from the comparison results in Section 6.2.3, where the explicit alignment model improved the alignment performance over the implicit model on both the basic and the structural representations of conversation segments, here there is no significant difference between the entailment performances of the two alignment models based on the basic representation of conversation segments (as shown in Figure 6.7(a)). This provides an evidence for the phenomenon previously found by other researchers [60], that a better alignment performance does not necessarily transfer to



(a) Based on basic representation

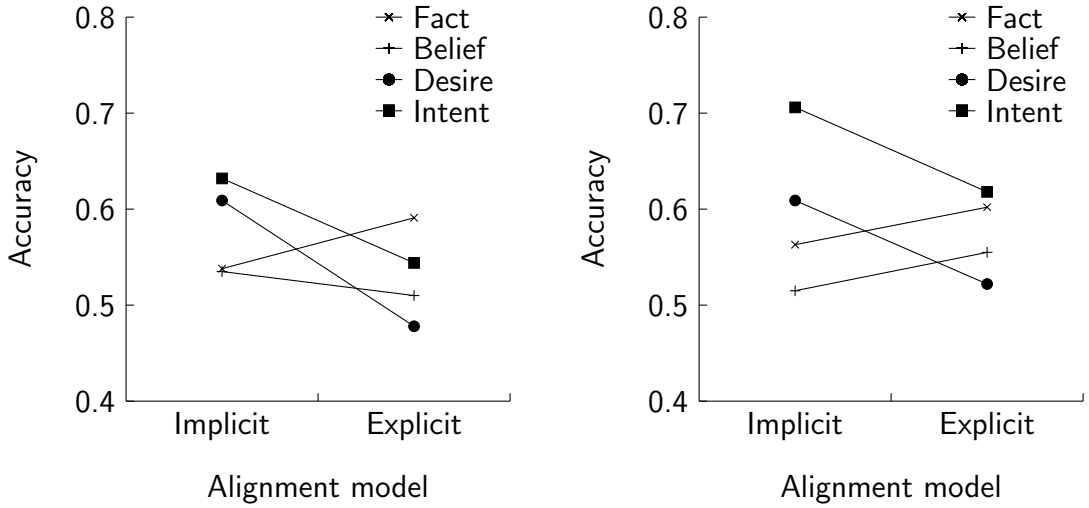
(b) Based on structural representation

Figure 6.7: Effect of different alignment models on entailment performance

better inference performance in entailment tasks.

However, when conversation structures are incorporated in the representation of conversation segments, the explicit alignment model has made certain improvement over the implicit alignment model (as in Figure 6.7(b)). Though none of these improvements are statistically significant, we may hypothetically extend our observation from Section 6.3.3 to the situation here. That is, in alignment models, the advantage of the explicit modeling of long distance relationship over the implicit modeling is also dependent on the incorporation of conversation structures in the conversation representation.

We further break down the comparison results for the two configurations in Figure 6.7(b) by different types of hypotheses, and the results are shown in Figure 6.8. We can see that in both Figure 6.8(a) and Figure 6.8(b), the explicit alignment model improves the entailment performance for the hypothesis type of *fact* compared to the implicit alignment model (in Figure 6.8(a) the improvement is statistically significant by McNemar's test, $p < 0.05$). This is consistent with what we found in Section 6.2.3: the advantage of explicit modeling of long distance relationship in alignment model



(a) With structural representation and implicit inference model

(b) With structural representation and explicit inference model

Figure 6.8: Effect of different alignment models for different hypothesis types

is most noticeable for *fact* hypotheses.

On the other hand, Figure 6.8 also demonstrates the reason why the explicit alignment model did not bring significant improvement to entailment performance in Figure 6.7 – it decreases the entailment performance for hypotheses other than the *fact* type. So why does the explicit alignment model improve the alignment performance for all hypothesis types in Section 6.2.3 but decrease the entailment performance for certain subsets here? The cause can be illustrated by the follow example:

Premise:

B: Well, don't you think a lot of that is diet too?

A: and, a lot of that is diet. That's true.

Hypothesis:

A agrees that a lot of health has to do with diet.

This is a true entailment (assuming *that* in the premise is resolved to *health*). However, our current entailment system was not able to identify it correctly, because

the verb phrase *has to do* in the hypothesis has no alignment in the premise. Actually, the recognition that *has to do* is just a way of representing an arbitrary relationship requires the knowledge of paraphrasing. Nonetheless, our implicit alignment mistakenly aligns the hypothesis term *do* with the premise term *is* (both occurrences), which makes the inference model predict a positive entailment (which is correct). Since the explicit alignment model corrects this alignment mistake, the entailment cannot be recognized. This is another evidence to show that a better alignment performance does not necessarily mean a better entailment performance.

Chapter 7

Discussions

This thesis provides a first step in the research on conversation entailment. The best configuration of our system achieves 37.5% precision and 52.6% recall on the verb alignment task (which constitute 43.8% f-measure as in Figure 6.2(b)), and 58.7% accuracy on the entailment prediction task (as evaluated in Section 6.3.3). Although this is a significant improvement compared to the baseline system for textual entailment, a better performance is desirable.

In this chapter we identify several issues faced by the current system and discuss potential improvement.

7.1 Cross-validation

The size of data used in our investigation is very small. Currently we have 291 entailment examples for training and 584 entailment examples for testing. This affects both learning of a reliable model and evaluation of the model performance. To better use our limited amount of data, we conducted cross-validation evaluations, utilizing part of the test data set for training.

The methodology for our cross-validation experiments is a modified version of

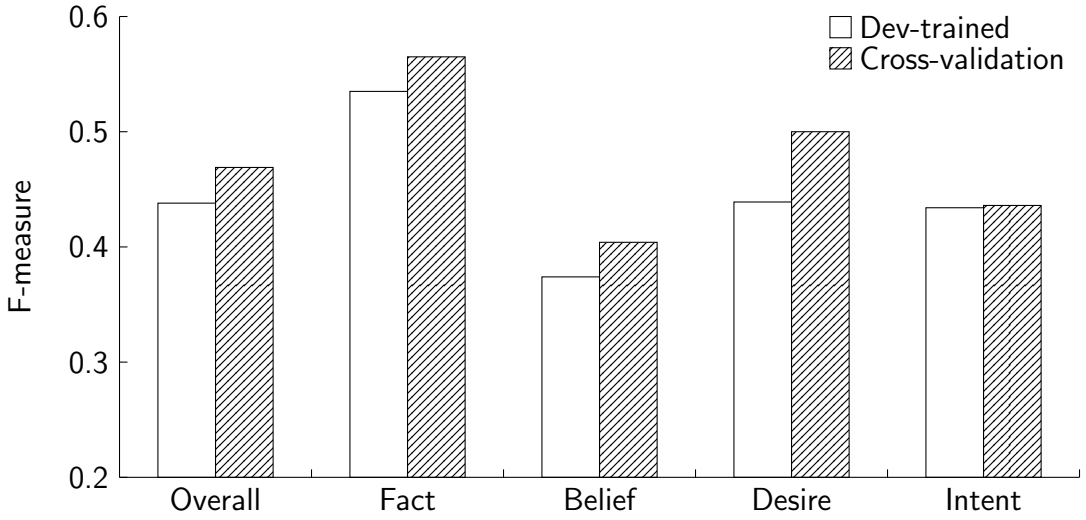


Figure 7.1: Comparing the cross-validation model and the model learned from development data for verb alignment results

leave-one-out evaluation. The entailment examples to be evaluated are the 584 examples in the test set, such that the evaluation results can be compared to our previous experiments. However, when evaluating each example in the test set, we use entailment models trained from the 291 development examples together with the 583 examples in the rest of the test set. This gives us 874 examples for each round of model training.

We conducted the cross-validation experiment for both the alignment model and the inference model, and we evaluate the results on both the verb alignment task and the entailment prediction task.

Figure 7.1 shows a comparison between the evaluation results on verb alignment produced by the alignment model with cross-validation, and the evaluation results produced by the alignment model trained from the development data only (*Dev-trained*, which comes from Figure 6.2(b)). Other configurations are the same for the two evaluations (structural conversation representation and explicit modeling of long distance relationship). Without any surprise, for the entire test set and for most of the hypothesis types, the cross-validation model achieves better performance than

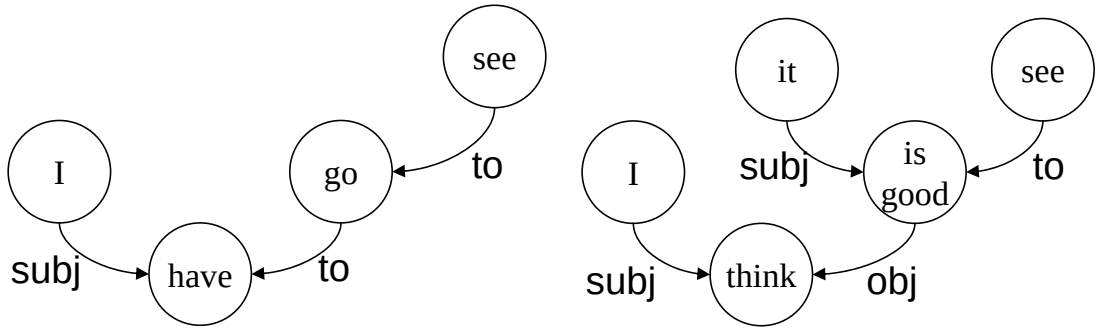
the model learned from development data only.

To study the cross-validation on the inference model, we conducted multiple experiments based on different alignment results produced by various alignment models (including the above results produced by cross-validation), and the best accuracy on entailment prediction is 58.9%. Compared to the previous result by inference model learned only from development data, 58.7%, there is no significant difference. This illustrates that (1) better alignment results do not necessarily lead to better entailment results, which is again consistent with the findings by MacCartney et al. [60] and by our previous experiments in Section 6.4.2; and (2) the performance of the inference model cannot be improved by simply feeding more training data. Better semantic representation in the models become critical.

7.2 Semantics

In Section 6.1.2 we mentioned there are different levels of modeling long distance relationship between language constituents using the pattern of path connecting them. Our current modeling of long distance relationship is on a relatively abstract level. That is, we model the long distance relationship between two language constituents by the types of nodes ($N/V/U$) and directions of edges ($\rightarrow/\leftarrow$) connecting them in a dependency structure. While such modeling captures some shallow semantics of the relationship between language constituents, it is very general and thus insufficient to differentiate specific semantic relations.

Consider two natural language statements "*I have to go see ...*" and "*I think it's good to see ...*". Figure 7.2(a) and Figure 7.2(b) show the dependency structures of these two statements respectively. In both figures, the long distance relationship



(a) *I have to go see ...* (b) *I think it's good to see ...*

Figure 7.2: The dependency structures for examples of shallow semantic modeling

between *see* and I is modeled as

$$V \rightarrow V \rightarrow V \leftarrow N$$

according to the current explicit modeling of LDR.

Therefore, an alignment model learned from the first example may recognize a verb-subject relationship represented by $V \rightarrow V \rightarrow V \leftarrow N$, since I is the logical subject of *see* in that example. However, when the alignment model is applied to the second example, such recognition becomes incorrect because I is not the logical subject of *see* in "I think it's good to see ...".

Similar problem also happens in inference models. For example, in statements "*A comes from Michigan*" and "*A went to Michigan*", the relations between A and *Michigan* are both modeled as $N \rightarrow V \leftarrow N$. If both statements are associated with the same hypothesis containing a relational clause $from(A, Michigan)$, an inference model learned from the first instance would recognize that $N \rightarrow V \leftarrow N$ entails the relation $from(\cdot, \cdot)$ (because *A comes from Michigan* entails $from(A, Michigan)$). But when the model is applied to the second instance, it will make a wrong prediction because in that case $N \rightarrow V \leftarrow N$ does not entail $from(\cdot, \cdot)$.

These problems can be resolved by adding more semantic information into the

explicit modeling of long distance relationship (using more detailed patterns to represent them). For example, in Figure 7.2(a), the relationship between *see* and I can be modeled as $V \xrightarrow{to} V \xrightarrow{to} V \xleftarrow{subject} I$, which is different from the relationship between *see* and I in Figure 7.2(b), $V \xrightarrow{to} V \xrightarrow{object} V \xleftarrow{subject} I$. And the relationship between A and *Michigan* in "*A comes from Michigan*" is $N \xrightarrow{from} V \xleftarrow{subject} N$, while in "*A went to Michigan*" it is $N \xrightarrow{to} V \xleftarrow{subject} N$. However, such finer grained models are not likely to be generalized well to new examples given our limited amount of data. Again, more training data will play an important role here.

Besides the semantic modeling of long distance relationship, our current system is also insufficient in modeling some types of lexical semantics, e.g., word sense and antonyms. Consider the verbs *see* and *hear*, in most cases they should not be aligned together, since they represent different types of actions. However, one of the senses for *see* in WordNet [64] is "get to know or become aware of", and so is *hear*. Therefore, without the capability of word sense disambiguation, our system frequently aligns these two terms together.

The importance of antonym modeling can be illustrated by the following example:

Premise:

A: Do yo-, are you on a reg-, regular exercise program right now?
B: Yes, and I hate it.

Hypothesis:

B doesn't like her exercise program.

A correct prediction of this true entailment would be to align the term *like* in the hypothesis to the term *hate* in the premise, and then recognize that the antonym of *hate* (*like*) plus a negative modifier *doesn't* has the same meaning as the original term *hate*. However, the polarity check module in our current system only checks for the number of negative modifiers but not the polarities of the verbs themselves. As the

verb *like* has one negative modifier and the verb *hate* has none, the polarity check module identifies *hate* and *doesn't like* as different polarities, and thus mistakenly predict the entailment to be false.

Lexical semantics have been extensively studied by other researchers [63, 68]. Tools and knowledge bases in this area should be utilized in the entailment system to improve their performance.

7.3 Pragmatics

The most important pragmatic features in conversation entailment are *ellipsis*, *pronoun usage*, and *conversation implicature*.

7.3.1 Ellipsis

In Section 5.1.3 we have summarized some unique features frequently seen in conversations. One of them is ellipsis. Ellipsis in conversations can be particularly challenging to both the alignment and the inference models. For example:

Premise:

A: Did you go to college?
B: I'm going right now.

Hypothesis:

B is going to college.

In the conversation utterance of speaker B, the object of the verb *going* is omitted, which is actually the term *college* in speaker A's utterance. Such a relationship between the verb *going* and its omitted object *college* needs to be recognized in the alignment model in order to align the hypothesis term *going* to the term *going* in

speaker B's utterance, and needs to be recognized in the inference model in order to infer the relational clause $to(going, college)$.

7.3.2 Pronoun Usage

In Table 4.1 we have seen a conversation entailment example with special pronoun usage:

Premise:

A: Sometimes unexpected meetings or a client would come in and would want to see you.

B: Right.

Hypothesis:

Sometimes a client wants to see B.

In this example the pronoun *you* in speaker A's utterance does not refer to speaker B, but has a meaning of speaker A him/herself. In other cases, a pronoun can refer to a general concept, for example:

Premise:

A: Um, matter of fact in the United States we used to have extended families.

B: Uh-huh.

Hypothesis:

A used to have extended families.

where the pronoun *we* in speaker A's utterance is a reference to the general concept of *people in United States*, while not necessarily involve speaker A him/herself.

In both of these cases, the generic or rhetorical usages of pronouns pose special challenges to the correct entailment prediction for the hypotheses.

7.3.3 Conversation Implicature

In Section 4.3.3 we have pointed out that conversation entailment is a quite challenging task, for the human annotators could not reach agreements on the entailment decisions for a considerable number of examples. We attribute some of the disagreements to different understandings of conversation implicature among the annotators.

Here we have another example, which did not cause that much trouble for the human annotators, but is still challenging to our entailment system due to the difficulty in the recognition of conversation implicature.

Premise:

A: While learning aerobics, you can just trust someone else.

Hypothesis:

A trusts her aerobics instructor.

In this example, in order to recognize that the hypothesis is a true entailment from the conversation segment, the system has to recognize in the conversation that *someone else* implies a person who teaches aerobics.

7.4 Knowledge

As discussed in Section 2.1.4, a key factor that affects the performance of an entailment system is the amount of knowledge available to the system. In our entailment system, all of its components (clause representation, alignment model, inference model) contains certain kinds of knowledge. To some extent, the limitations of the current system in semantics and pragmatics (as discussed in Section 7.2 and Section 7.3) are essentially due to lack of knowledge.

In this section we discuss about two more kinds of knowledge that are missing from our system.

7.4.1 Paraphrase

In Section 6.4.2 we have seen an entailment example that requires the knowledge of paraphrasing. Here is another one:

Premise:

A: My TV viewing started sort of mid-sixties when I was really little.
B: I see.

Hypothesis:

At mid-sixties, A was a small child.

Our entailment system fails to recognize that this is a true entailment, because it cannot find the alignment for the hypothesis term *child* in the premise. While a knowledgeable entailment system would recognize that *was a small child* is in fact another way of saying *was really little*.

The knowledge of paraphrases has been accumulated from large linguistic co-pra [55, 78, 86], and has been applied to the text entailment task by other researchers [27]. However, most of these efforts limit their acquisition and application of paraphrases to binary representations, i.e., there are two variables in the paraphrases (e.g., *X prevents Y* $\rightarrow$ *X provide protection against Y*). However, as seen in the example above, many times unary paraphrases are also useful in entailment recognition (e.g., *X was really little* $\rightarrow$ *X was a small child*). Recently there is work on the acquisition of such paraphrases [85]. The applications of this type of paraphrases on the entailment task will be interesting to investigate.

7.4.2 World Knowledge

The importance of world knowledge in the conversation entailment task can be demonstrated by the following example:

Premise:

A: I use my credit card a great deal, um, for groceries.

Hypothesis:

Hypothesis: A does grocery shopping.

The necessary (and sufficient) knowledge used to recognize this entailment is that a *credit card* is used for *shopping*. Unfortunately, such knowledge is missing in our system, such that the term *shopping* in the hypothesis becomes a new entity, for which it cannot find an alignment in the premise. As a result, the system is unable to predict this is a true entailment.

7.5 Efficiency

Generally speaking, our entailment system is designed for offline processing. The conversation segments and the hypotheses are given in batches. Thus efficiency has not been a main focus in our development. Here we provide some general discussion on the efficiency issue should our entailment system become a part of online application.

There are mainly three components in our system, the decomposition model, the alignment model, and the inference model.

The decomposition model works upon decomposition rules in Appendix A. Its efficiency is dependent on its input – syntactic parse trees and the number of decomposition rules. Overall speaking, the complexity of the decomposition model is proportional to the size (number of internal nodes) of the syntactic trees. However, when it comes down to a particular substructure in the syntactic tree (e.g. $S \rightarrow NP\ VP$), its complexity varies. It is simpler if there is a matching rule in the rule set for the syntactic substructure (e.g., $S \rightarrow NP\ VP$), in this case the processing time is constant for that substructure. However, when there is not a match for the substructure (e.g. $S \rightarrow NP\ VP\ PP$), as described in Appendix A, the system will try to search for a

rule to reduce this substructure (e.g., use $VP \rightarrow VP\ PP$ to reduce $S \rightarrow NP\ VP\ PP$ to $S \rightarrow NP\ VP$). This is a recursive process. In the case that a single span in the syntactic tree is very large (i.e., one parent has many children on the same layer), the process can go very slow. Fortunately, this rarely happens on the conversation data (since utterance lengths are short).

The complexities of the alignment and inference models can be further divided into three parts: feature calculation, model building, and model application.

The calculations of binary features (*string equality*, *stemmed equality*, *acronym equality*, *named entity equality*, and *verb be identification*) are trivial. The calculation of WordNet similarity can be efficient, as long as the relevant probabilities of each WordNet class are pre-calculated and stored. The calculation of distributional similarity needs to get the document count for particular terms in a large text corpus. We use the Lemur retrieval engine[1] to handle efficient query search.

The calculations of LDR features (*subject consistency*, *object consistency*, and the feature for the relational inference model) are dependent on the sizes of alignments, i.e., for each hypothesis term how many premise terms it is aligned to. This size statistics depend on the alignment model used and (for implicit alignment model) the output threshold for the logistic regression model. For the implicit alignment model with threshold 0.7, on average a hypothesis term is aligned to 1.57 premise terms; and for explicit model, a hypothesis term is aligned to 1.55 premise terms on average. In either case, the alignment size results in relatively low complexity.

After the premise terms are selected, the system needs to calculate the distance (for implicit LDR models) or the pattern of path (for explicit LDR models) between the selected terms. It turns out they are the same problem, with an efficient solution of breadth-first-search on a dependency graph. The complexity of such searching is bounded by the number of vertices in the graph, or the number of terms in the clause

[1] `http://www.lemurproject.org/`

representation of a premise. Statistics on our test data show that the average number of terms in a conversation segment is 31.3.

The complexities of model building and model application depend on the statistical model used. For logistic regression model, the model application can be done in constant time, and the main complexity lies in model building. This is an iterative procedure (a Quasi-Newton search method). We used the toolkit provided by Weka[2], which is an implementation of the algorithm given by le Cessie and van Houwelingen [53], known to converge quickly.

For k-nearest neighbour model, the "model building" is just storing training instances. So the complexity mainly lies in model application, which in our current implementation is proportional to the number of training instances for each test instance. Currently our (kNN) models are trained from the development set of 291 entailment examples. On average each entailment example has 32.4 premise terms and 5.38 hypothesis terms, so the average number of alignment instances (potential pairings of terms between each hypothesis and premise) is roughly around 174 for each entailment example, and about 5×10^4 for the whole development set. Moreover, the total number of relational clauses in the hypotheses of the development set is 727. So for both the alignment model and the inference model, our current implementations of kNN models have acceptable efficiency. However, if a larger set of training data become available, speedy search techniques for k-nearest neighbours should be considered (e.g., spatial indices).

[2] `http://www.cs.waikato.ac.nz/ml/weka/`

Chapter 8

Conclusion and Future Work

Currently there is no data or published studies that address the problem of conversation entailment. This thesis is one of the first studies that investigate this problem. In this chapter, we summarize our contributions and future work.

8.1 Contributions

This thesis has made the following contributions.

1. Systems and computational models addressing the conversation entailment problem.

   We developed a probabilistic framework for the entailment problem. The framework is based on the representation of dependency structures of language. The overall entailment prediction depends on the entailment relations between substructures. For conversation entailment, we incorporated conversation modeling (e.g., dialogue acts, conversation participants, and conversation discourse) into the dependency structures. The modeling of conversation structures has been shown effective and has contributed to a significant improvement (an absolute difference of 4.8%) in system performance. Especially, it is critical for predicting

the entailment of *intent* type of hypotheses (with an absolute improvement of 25%).

2. A systematic investigation on the roles and interactions of different models and representations in the overall entailment prediction.

   We experimented with several alignment and inference models in the conversation entailment system, investigating different approaches of shallow semantic modeling. Our studies have shown that the explicit modeling of long distance relationship based on the path between two language constituents is useful in both the alignment and the inference models. It improved the entailment performance by 3.9% on the test data compared to the implicit modeling of long distance relationship (based on distance). Specifically, its effect in the alignment model is more prominent for the *fact*, *belief*, and *desire* types of hypotheses, while its effect in the inference model is the most prominent for *intent* hypotheses. In addition, the effect of explicit modeling of long distance relationship is largely dependent on the presence of structural information in conversation representations. Similarly, the modeling of conversation structure is more effective when the computational models incorporate shallow semantic information (using explicit modeling of long distance relationship).

3. A data corpus of conversation entailment examples to facilitate the initial investigation on conversation entailment.

   We collected 1096 conversation entailment examples. Each example consists of a segment of conversation discourse and a hypothesis statement. The data set used in this thesis includes 875 examples with at least 75% agreement among five annotators. This data set is made available at `http://links.cse.msu.edu:8000/lair/projects/conversationentailment_data.html`. Although a larger data set is preferable, the small collection of data resulted from this thesis

supports initial investigation and evaluation on conversation entailment.

8.2 Future Work

Conversation entailment is a challenging problem. This thesis only presents an initial investigation. More systematic and in-depth studies are required to further understand the nature of the problem and develop better technologies.

Data. The availability of relevant data is always a major issue in language related research. Although our current data enabled us to start an initial investigation on conversation entailment, its small size poses significant limitations on technology development and evaluation. A more systematical approach to collect and create a large set of data is crucial. One possible future direction is to develop innovative community-based approach (e.g., through web) for data collection. Annotations based on Mechanical Turk can also be pursued.

Semantics and Pragmatics. As discussed in Sections 7.2, 7.3, and 7.4, our semantic modeling in the entailment system is pretty shallow. Our clause representation for both the conversation segment and the hypothesis statement is mainly syntactic-driven (based on dependency structure). As more techniques in semantic processing (e.g., semantic role) become available, representation should capture deeper semantics. The models should also address pragmatics (e.g., conversation implicature) and incorporate more world knowledge.

Applications. Finally, as the technology in conversation entailment is developed, its applications in NLP problems should be explored. Example applications include information extraction, question answering, summarization from conversation scripts,

and modeling of conversation participants. These applications may provide new insights on the nature of the conversation entailment problem and its potential solutions.

Appendices

Appendix A

Syntactic Decomposition Rules

This appendix lists the rules used for syntactic decomposition (Section 3.1.1).

Each decomposition rule is built upon a grammar rule (e.g. $S \rightarrow NP\ VP$) and has two parts. The first part selects the head for each syntactic constituent. For example, for $S \rightarrow NP\ VP$ we define the head of VP to be the head of S. Since VP is the second child of S, we represent its head as h_2. The head of VP is then obtained recursively from rules spanning VP (e.g., $VP \rightarrow VP\ NP$). The head of a leaf constituent is a term representing this constituent (e.g., for $NNP \rightarrow John$ we have $x_1 = John$). It is possible that a head is the concatenation of multiple children (e.g., for $NP \rightarrow NNP\ NNP$ the head of NP is derived by $h_1 h_2$), in which case we use a single term to represent the concatenated entity. It is also possible that a constituent has two heads (e.g. for $NP \rightarrow NP\ CONJP\ NP$ we have h_1, h_3). The difference between these two notions should be noted.

The second part generates property or relational clauses from the syntactic substructure. For example, for $VP \rightarrow ADVP\ VP$ we generate a property clause $h_1(h_2)$ which means the head of second child (VP) has a property described by the head of the first child ($ADVP$). For $S \rightarrow NP\ VP$ we generate a relational clause $subj(h_2, h_1)$ which means the subject of the head of second child (VP) is the head of first child

(NP). It is possible for a single grammar rule to generate multiple clauses.

Grammar rules that are not included in this list can usually be reduced into two or more basic rules. For example, for $S \rightarrow NP\ VP\ PP$, it can be seen as a combination of $S \rightarrow NP\ VP$ and $VP \rightarrow VP\ PP$. So we first apply the rule $VP \rightarrow VP\ PP$ to the last two children of $S \rightarrow NP\ VP\ PP$, and reduce it to $S \rightarrow NP\ VP$, which have a second applicable rule in our rule set. In this way we can deal with infinite number of syntactic structures. For example, for the rules

$$NP \rightarrow NNP\ NNP$$
$$NP \rightarrow NNP\ NNP\ NNP$$
$$NP \rightarrow NNP\ NNP\ NNP\ NNP$$
$$\ldots$$

There could be arbitrary number of NNP nodes in the child list of NP. So we define the following rules

$$\begin{aligned} nnp &\rightarrow NNP \\ nnp &\rightarrow NNP\ nnp \\ NP &\rightarrow nnp \end{aligned} \tag{A.1}$$

Such that the reduction of rule $NP \rightarrow NNP\ NNP\ NNP\ NNP$ can be

$$NP \rightarrow NNP\ NNP\ NNP\ nnp$$
$$NP \rightarrow NNP\ NNP\ nnp$$
$$NP \rightarrow NNP\ nnp$$
$$NP \rightarrow nnp$$

As we can see, the grammar $NP \rightarrow NNP \ \ldots \ NNP$ with arbitrary number of NNP's can always be reduced into a combination of the three basic rules defined in (A.1) (note here the syntactic constituents are case-sensitive, with lower case constituents serving as intermediate layer, not occurring in an original grammar rule).

The full set of decomposition rules that we use is listed in Table A.1.

Table A.1: Rules for syntactic decomposition

Grammar rule	Head(s)	Clause(s)
nnp → NNP	h_1	
nnp → NNPS	h_1	
nnp → FW	h_1	
nnp → NNP nnp	$h_1 h_2$	
nnp → FW nnp	$h_1 h_2$	
np → nnp	h_1	
np → QP	h_1	
np → NN	h_1	
np → NNS	h_1	
np → VBG	h_1	
NP → np	h_1	
NP → ADJP	h_1	
NP → DT ADJP	$h_1 h_2$	
NP → NP POS	h_1	
NP → PRP	h_1	
NP → PRP$	h_1	
NP → DT	h_1	
NP → EX	h_1	
NP → DT NP	h_2	

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
NP → NP NP	h_2	$modifier(h_2, h_1)$
NP → attr NP	h_2	$h_1(h_2)$
NP → NP attr	h_1	$h_2(h_1)$
NP → NP PP	h_1	$preposition(h_1, h_2)$
NP → NP SBAR	h_1	$modifier(h_1, h_2)$
NP → NP , NP	h_1	$is(h_1, h_3)$
NP → NP PRN	h_1	$is(h_1, h_2)$
NP → “ NP	h_2	
NP → NP ”	h_1	
NP → NP ,	h_1	
NP → NP .	h_1	
NP → -LRB- NP -RRB-	h_2	
npcc → NP CONJP NP	h_1, h_3	
npcc → NP CC PRN NP	h_1, h_4	
npcc → NP , npcc	h_1, h_3	
NP → npcc	h_1	
NP → VBG NP	h_1, h_2	$modifier(h_2, h_1)$, $object(h_1, h_2)$
NP → NP S	h_1	$subject(h_2, h_1)$
NP → NP : NP	h_1	$is(h_1, h_3)$
NP → ADVP NP	h_2	$h_1(h_2)$
NP → ADVP npcc	h_2	$h_1(h_2)$
NP → NP ADVP	h_1	$h_2(h_1)$
NP → ADVP	h_1	
NP → NP : NP :	h_1	$is(h_1, h_3)$

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
NP → CC NP CC NP	h_2, h_4	
NP → PDT NP	h_2	$h_1(h_2)$
NP → NX	h_1	
DT → ADVP DT	$h_1 h_2$	
attr → ADJP	h_1	
attr → VP	h_1	
ADJP → JJ	h_1	
ADJP → JJR	h_1	
ADJP → JJS	h_1	
ADJP → ADVP ADJP	$h_1 h_2$	
ADJP → “ ADJP ”	h_2	
ADJP → -LRB- ADJP -RRB-	h_2	
ADJP → ADJP ,	h_1	
ADJP → ADJP CC ADJP	h_1, h_3	
ADJP → ADJP PRN	h_1, h_2	
ADJP → ADJP PP	$h_1 h_2$	
ADJP → ADJP S	$h_1 h_2$	
ADJP → NP ADJP	$h_1 h_2$	
ADJP → ADJP SBAR	$h_1 h_2$	
ADJP → RB	h_1	
ADJP → UCP	h_1	
ADVP → RP	h_1	
ADVP → RB	h_1	
ADVP → RBR	h_1	

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
ADVP → RBS	h_1	
ADVP → ADVP ADVP	h_1, h_2	
ADVP → NP RB	$h_1 h_2$	
ADVP → ADVP ,	h_1	
ADVP → IN	h_1	
ADVP → ADVP CONJP ADVP	h_1, h_3	
ADVP → ADVP PP	$h_1 h_2$	
QP → CD	h_1	
QP → CD QP	$h_1 h_2$	
QP → QP TO QP	$h_1 h_2 h_3$	
QP → $ QP	$h_1 h_2$	
QP → JJ IN QP	$h_1 h_2 h_3$	
QP → IN JJS QP	$h_1 h_2 h_3$	
QP → RB QP	$h_1 h_2$	
QP → QP CONJP QP	h_1, h_3	
CONJP → CC	h_1	
CONJP → CC ADVP	$h_1 h_2$	
CONJP → CC , ADVP	$h_1 h_3$	
PP → IN NP	$h_1 h_2$	
PP → TO NP	$h_1 h_2$	
PP → IN S	$h_1 h_2$	
PP → IN SBAR	$h_1 h_2$	
PP → vb NP	$h_1 h_2$	
PP → vb PP	$h_1 h_2$	

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
PP → vb SBAR	$h_1 h_2$	
PP → : PP :	h_2	
PP → PP ,	h_1	
PP → ADVP PP	$h_1 h_2$	
PP → ADJP PP	$h_1 h_2$	
PP → IN PP	$h_1 h_2$	
PP → IN ADJP	$h_1 h_2$	
PP → IN ADVP	$h_1 h_2$	
prn → S	h_1	
prn → SBARQ	h_1	
prn → prn ,	h_1	
prn → , prn	h_2	
prn → . prn	h_2	
PRN → prn	h_1	
PRN → -LRB- NP -RRB-	h_2	
PRN → -LRB- CC NP -RRB-	$h_2 h_3$	
vb → VB	h_1	
vb → VBD	h_1	
vb → VBN	h_1	
vb → VBZ	h_1	
vb → VBP	h_1	
vb → VBG	h_1	
vb → BES	h_1	
vb → HVS	h_1	

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
VP → vb	h_1	
VP → vb VP	$h_1 h_2$	
VP → MD VP	$h_1 h_2$	
VP → VP NP	h_1	$object(h_1, h_2)$
VP → VP : NP	h_1	$object(h_1, h_3)$
VP → TO VP	h_2	
VP → ADVP VP	h_2	$h_1(h_2)$
VP → VP ADVP	h_1	$h_2(h_1)$
VP → VP PRT	h_1	$h_2(h_1)$
VP → VP PP	h_1	$preposition(h_1, h_2)$
VP → PP VP	h_2	$preposition(h_2, h_1)$
VP → VP S	h_1	$adverbial(h_1, h_2)$
VP → VP SBAR	h_1	$object(h_1, h_2)$, $adverbial(h_1, h_2)$
VP → vb ADJP	$h_1 h_2$	
VP → VP ,	h_1	
VP → , VP	h_2	
vpcc → VP CONJP VP	h_1, h_3	
vpcc → VP , vpcc	h_1, h_3	
vpcc → VP CONJP vpcc	h_1, h_3	
VP → vpcc	h_1	
S → S ,	h_1	
S → S .	h_1	
S → “ S	h_2	
S → S ”	h_1	

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
S → VP	h_1	
S → NP VP	h_2	$subject(h_2, h_1)$
S → PP S	h_2	$preposition(h_2, h_1)$
S → ADVP S	h_2	$h_1(h_2)$
S → SBAR , S	h_3	$adverbial(h_3, h_1)$
S → S CC S	h_1, h_3	
S → CC S	h_2	
S → CC , S	h_3	
S → PRN S	h_2	
S → NP ADJP	$h_1 h_2$	$h_2(h_1)$
S → S S	h_1, h_2	
S → NP	h_1	
S → CONJP S	$h_1 h_2$	
SBAR → S	h_1	
SBAR → CC S	h_2	
SBAR → WHNP S	h_2	
SBAR → WHADVP S	h_2	
SBAR → WHPP S	h_2	
SBAR → IN S	$h_1 h_2$	
SBAR → RB IN S	$h_1 h_2 h_3$	
SBARQ → WHNP SQ .	h_2	$subject(h_2, h_1)$
SBARQ → WHADVP SQ .	h_2	$subject(h_2, h_1)$
WHNP → WP	h_1	
WHNP → WDT	h_1	

Table A.1: (continued)

Grammar rule	Head(s)	Clause(s)
WHNP → WHNP PP	$h_1 h_2$	
WHADVP→ WRB	h_1	
SQ → S	h_1	
PRT → RP	h_1	
PRT → RB	h_1	

Appendix B

List of Dialogue Acts

Table B.1 lists the 69 dialogue act labels used by the annotation system of Switchboard dialogue corpus [38].

Table B.1: The dialogue act labels used by Switchboard annotation system

q	question
s	statement
b	backchannel/backwards-looking
f	forward-looking
a	agreements
%	indeterminate, interrupted, or contains just a floor holder
(^u	unrelated response (first utterance is not response to previous q)
*	comment (followed by *[[comment...]] after transcription to explain)
+	continued from previous by same speaker
@,o@,+@	incorrect transcription (can add comment to specify problem further)

Table B.1: (continued)

^2	collaborative completion
^c	about-communication
^d	declarative question (question asked like a structural statement)
^e	[on statements] elaborated reply to *yes-no*-question
^g	tag question (question asked like a structural statement with a question tag at end)
^h	hold (often but not always after a question) (*Let me think*) (question in response to a question)
^m	mimic other
^q	quotation
^r	repeat self
^t	about-task
aap	accept-part
ad	action-directive (*Go ahead. We could go back to television shows*)
aa	accept (*Ok. I agree*)
am	maybe
ar	reject (*no*)
arp	reject-part
b	default agreement or continuer (*uh-huh*, *right*, *yeah*)
b^m	repeat-phrase
ba	assessment/appreciation (*I can imagine*)
bc	correct-misspeaking
bd	downplaying-response-to-sympathy/compliments (*That's all right. That happens*)

Table B.1: (continued)

bf	reformulate/summarize; paraphrase/summary of other's utterance (as opposed to a mimic)
bh	rhetorical question continuer (*Oh really?*)
bk	acknowledge-answer (*Oh, okay*)
br	signal-non-understanding (request for repeat)
br^m	signal-non-understanding via mimic
br^c	non-understanding due to problems with phone line
by	sympathetic comment (*I'm sorry to hear about that*)
cc	commit
co	offer
fa	apology (*Apologies*) (this is not the *I'm sorry* of sympathy which is by)
fc	conventional-closing
fe	exclamation (*Ouch*)
fo	other-forward-function
fp	conventional-opening
ft	thanks (*Thank you*)
fw	welcome (*You're welcome*)
fx	explicit-performative (*you're filed*)
na	a descriptive/narrative statement which acts as an affirmative answer to a question
nd	answer dispreferred (*Well...*)
ng	a descriptive/narrative statement which acts as a negative answer to a question
nn	no or variations (only)

Table B.1: (continued)

no	a response to a question that is neither affirmative nor negative (often *I don't know*)
ny	yes or variations (only)
o	other
oo	open-option (*We could have lamb or chicken*)
qh	rhetorical question
qo	open ended question
qr	alternative (*or*) question
qrr	an *or*-question clause tacked onto a *yes-no*-question
qw	*wh*-question
qy	*yes-no*-question
sd	descriptive and/or narrative (listener has no basis to dispute)
sv	viewpoint, from personal opinions to proposed general facts (listener could have basis to dispute)
t1	self-talk
t3	third-party-talk
x	nonspeech

The tags in Table B.1 were used in combination in annotating the Switchboard conversations. Thus a total number of 226 combined labels were created. After that, they removed the tag combinations that occurred infrequently, removed the secondary carat-dimensions (^2, ^g, ^m, ^r, ^e, ^q, ^d, but with some exceptions [38]), and grouped together some tags that had very little training data. This resulted in 42 classes of dialogue acts. The mapping between the dialogue act classes and the original

tags are in Table B.2. These 42 acts were later summarized as a comprehensive list by Stolcke et al. [84]. In this thesis we also use the same set as the tagging system of dialogue acts.

Table B.2: The dialogue acts used in this thesis

Dialogue act	Tag	Example
Statement-non-opinion	sd	Me, I'm in the legal department.
Acknowledge (Backchannel)	b	Uh-huh.
Statement-opinion	sv	I think it's great.
Agree/Accept	aa	That's exactly it.
Abandoned or Turn-Exit	% -	So, -
Appreciation	ba	I can imagine.
Yes-No-Question	qy	Do you have to have any special training?
Non-verbal	x	[Laughter], [Throat_clearing]
Yes answers	ny	Yes.
Conventional-closing	fc	Well, it's been nice talking to you.
Wh-Question	qw	Well, how old are you?
No answers	nn	No.
Response Acknowledgement	bk	Oh, okay.
Hedge	h	I don't know if I'm making any sense or not.
Declarative Yes-No-Question	qy^d	So you can afford to get a house?
Other	o fo bc by fw	Well give me a break, you know.
Backchannel in question form	bh	Is that right?
Quotation	^q	You can't be pregnant and have cats.

Table B.2: (continued)

Dialogue act	Tag	Example
Summarize/reformulate	bf	Oh, you mean you switched schools for the kids.
Affirmative non-yes answers	na ny^e	It is.
Action-directive	ad	Why don't you go first?
Collaborative Completion	^2	Who aren't contributing.
Repeat-phrase	b^m	Oh, fajitas.
Open-Question	qo	How about you?
Rhetorical-Questions	qh	Who would steal a newspaper?
Hold before answer/agreement	^h	I'm drawing a blank.
Reject	ar	Well, no.
Negative non-no answers	ng nn^e	Uh, not a whole lot.
Signal-non-understanding	br	Excuse me?
Other answers	no	I don't know.
Conventional-opening	fp	How are you?
Or-Clause	qrr	or is it more of a company?
Dispreferred answers	arp nd	Well, not so much that.
3rd-party-talk	t3	My goodness, Diane, get down from there.
Offers, Options Commits	oo cc co	I'll have to check that out.
Self-talk	t1	What's the word I'm looking for.
Downplayer	bd	That's all right.
Maybe/Accept-part	aap am	Something like that.
Tag-Question	^g	Right?
Declarative Wh-Question	qw^d	You are what kind of buff?

Table B.2: (continued)

Dialogue act	Tag	Example
Apology	`fa`	I'm sorry.
Thanking	`ft`	Hey thanks a lot.

Bibliography

[1] E. Akhmatova. Textual entailment resolution via atomic propositions. In *Proceedings of the PASCAL RTE Challenge Workshop*, 2005.

[2] J. Allen. *Natural language understanding*. The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, USA, 1995.

[3] J. Allen and M. Core. *Draft of DAMSL: Dialog Act Markup in Several Layers*, 1997.

[4] A. Anderson, M. Bader, E. Bard, E. Boyle, G. M. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson, and R. Weinert. The hcrc map task corpus. *Language and Speech*, 34:351–366, 1991. EN.

[5] J. L. Austin. *How to Do Things with Words*. Harvard University Press, Cambridge, MA, 1962.

[6] L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, March 1983.

[7] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

[8] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy, 2006.

[9] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[10] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini. The fifth pascal recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA, 2009.

[11] T. Bocklet, A. Maier, and E. Nöth. Age determination of children in preschool and primary school age with gmm-based supervectors and support vector machines/regression. In *TSD '08: Proceedings of the 11th international conference on Text, Speech and Dialogue*, pages 253–260, Berlin, Heidelberg, 2008. Springer-Verlag.

[12] J. Bos and K. Markert. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[13] C. Boulis and M. Ostendorf. A quantitative analysis of lexical differences between genders in telephone conversations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 435–442, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[14] C. Brockett. Aligning the rte 2006 corpus, 2007. Technical Report MSR-TR-2007-77.

[15] G. Carenini, R. T. Ng, and X. Zhou. Summarizing email conversations with clue words. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 91–100, New York, NY, USA, 2007. ACM.

[16] J. C. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. M. Post, D. Reidsma, and P. Wellner. The ami meeting corpus: A pre-announcement. In S. Renals and S. Bengio, editors, *Machine Learning for Multimodal Interaction, Second International Workshop, Edinburgh, UK*, volume 3869 of *Lecture Notes in Computer Science*, pages 28–39, Berlin, 2006. Springer Verlag.

[17] F. Chang, G. S. Dell, and K. Bock. Becoming syntactic. *Psychological Review*, 113(2):234–272, April 2006.

[18] Y.-W. Chen and C.-J. Lin. Combining svms with various feature selection strategies. In I. Guyon, M. Nikravesh, S. Gunn, and L. Zadeh, editors, *Feature Extraction*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 315–324. Springer Berlin / Heidelberg.

[19] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA, February 1988. Association for Computational Linguistics.

[20] J. Coates. *Language and gender: a reader*. Wiley-Blackwell, 1998.

[21] S. Cohen. A computerized scale for monitoring levels of agreement during a conversation. In *Proceedings of the 26th Penn Linguistics Colloquium*, 2002.

[22] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *PASCAL Challenges Workshop on Recognising Textual Entailment*, Southampton, U.K, 11 - 13 April 2005.

[23] C. C. David, D. Miller, and K. Walker. The fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings 4th International Conference on Language Resources and Evaluation*, pages 69–71, 2004.

[24] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC*, 2006.

[25] R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. An inference model for semantic entailment in natural language. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

[26] E. Dermataso and G. Kokkinakis. Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2):137–163, June 1999.

[27] G. Dinu and R. Wang. Inference rules and their application to recognizing textual entailment. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 211–219, Athens, Greece, March 2009. Association for Computational Linguistics.

[28] G. Doddington, A. Mitchell, M. Przybocki, and L. Ramshaw. The automatic content extraction (ace) program–tasks, data, and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, 2004.

[29] P. Eckert and S. McConnell-Ginet. *Language and gender*. Cambridge University Press, 2003.

[30] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[31] V. S. Ferreira and K. Bock. The functions of structural priming. *Language and Cognitive Processes*, 21(7-8):1011–1029, November 2006.

[32] A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi, and J. Stephan. Applying cogex to recognize textual entailment. In *Proceedings of the PASCAL RTE Challenge Workshop*, 2005.

[33] M. Galley. A skip-chain conditional random field for ranking meeting utterances by importance. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 364–372, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[34] M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 669–676, Barcelona, Spain, July 2004.

[35] N. Garera and D. Yarowsky. Modeling latent biographic attributes in conversational genres. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 710–718, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[36] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June 2007. Association for Computational Linguistics.

[37] D. Giampiccolo, H. T. Dang, B. Magnini, I. Dagan, E. Cabrio, and B. Dolan. The fourth pascal recognizing textual entailment challenge. In *Proceedings of the First Text Analysis Conference (TAC 2008)*, Gaithersburg, Maryland, USA, 2008.

[38] J. J. Godfrey and E. Holliman. *Switchboard-1 Release 2*. Linguistic Data Consortium, Philadelphia, 1997.

[39] D. Graff. *The AQUAINT Corpus of English News Text*. Linguistic Data Consortium, Philadelphia, 2002.

[40] A. Haghighi, A. Ng, and C. Manning. Robust textual inference via graph matching. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 387–394, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[41] Z. S. Harris. Distributional structure. In J. J. Katz, editor, *The Philosophy of Linguistics*, pages 26–47. Oxford University Press, Oxford, 1985.

[42] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain, July 1997. Association for Computational Linguistics.

[43] A. Hickl. Using discourse commitments to recognize textual entailment. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 337–344, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[44] A. Hickl and J. Bensley. A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176, Prague, June 2007. Association for Computational Linguistics.

[45] J. Hirschberg and D. Litman. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530, September 1993.

[46] J. R. Hobbs, M. E. Stickela, D. E. Appelta, and P. Martina. Interpretation as abduction. *Artificial Intelligence*, 63(1-2):69–142, October 1993.

[47] A. Iftene and A. Balahur-Dobrescu. Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 125–130, Prague, June 2007. Association for Computational Linguistics.

[48] V. Jijkoun and M. de Rijke. Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenge Workshop on Recognising Textual Entailment*, pages 73–76, 2005.

[49] H. Jing, N. Kambhatla, and S. Roukos. Extracting social networks and biographical facts from conversational speech transcripts. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 1040–1047, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[50] P. Kingsbury and M. Palmer. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Canary Islands, Spain, 2002.

[51] K. Kipper, H. T. Dang, and M. Palmer. Class-based construction of a verb lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 691–696. AAAI Press / The MIT Press, 2000.

[52] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[53] S. le Cessie and J. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[54] D. Lin. An information-theoretic definition of similarity. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[55] D. Lin and P. Pantel. Dirt - discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, pages 323–328, 2001.

[56] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the Twentieth International Conference on Machine Learning*, Washington, DC, August 2003.

[57] R. K. S. Macaulay. *Talk that counts: age, gender, and social class differences in discourse*. Oxford University Press US, 2005.

[58] B. MacCartney and C. D. Manning. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[59] B. MacCartney, T. Grenager, M.-C. de Marneffe, D. Cer, and C. D. Manning. Learning to recognize features of valid textual entailments. In *Proceedings of HLT-NAACL*, 2006.

[60] B. MacCartney, M. Galley, and C. D. Manning. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[61] M. P. Marcus, B. Santorini, M. A. Marcinkiewicz, and A. Taylor. *Treebank-3*. Linguistic Data Consortium, Philadelphia, 1999.

[62] S. Maskey and J. Hirschberg. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *INTERSPEECH-2005*, pages 621–624, 2005.

[63] D. McCarthy. Word sense disambiguation: An overview. *Language and Linguistics Compass*, 3(2):537–558, March 20 2009.

[64] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11): 39–41, 1995.

[65] D. Moldovan, C. Clark, S. Harabagiu, and S. Maiorano. Cogex: a logic prover for question answering. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 87–93, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[66] G. Murray and G. Carenini. Summarizing spoken and written conversations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 773–782, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[67] G. Murray, S. Renals, and J. Carletta. Extractive summarization of meeting recordings. In *in Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 593–596, 2005.

[68] R. Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2): 1–69, 2009.

[69] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*, 2000.

[70] S. Oviatt. Predicting spoken disfluencies during human-computer interaction. *Computer Speech and Language*, 9(1):19–35, 1995.

[71] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[72] M. J. Pickering and S. Garrod. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Science*, 27(2):169–190, 2004.

[73] A. Pomerantz. Agreeing and disagreeing with assessments: Some features of preferred/dispreferred turn shapes. In J. M. Atkinson and J. C. Heritage, editors, *Structures of Social Action*, pages 57–101. 1984.

[74] S. S. Pradhan, W. Ward, and J. H. Martin. Towards robust semantic role labeling. *Computational Linguistics*, 34(2):289–310, June 2008.

[75] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, January 1986.

[76] R. Raina, A. Y. Ng, and C. D. Manning. Robust textual inference via learning and abductive reasoning. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, 2005.

[77] D. Reitter and J. D. Moore. Predicting success in dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 808–815, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[78] S. Sekine. Automatic paraphrase discovery based on context and keywords between ne pairs. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, pages 80–87, 2005.

[79] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[80] S. Somasundaran, J. Ruppenhofer, and J. Wiebe. Detecting arguing and sentiment in meetings. Antwerp, September 2007.

[81] S. Somasundaran, J. Ruppenhofer, and J. Wiebe. Discourse level opinion relations: An annotation study. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 129–137, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[82] S. Somasundaran, J. Wiebe, and J. Ruppenhofer. Discourse level opinion interpretation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 801–808, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[83] S. Somasundaran, G. Namata, J. Wiebe, and L. Getoor. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 170–179, Singapore, August 2009. Association for Computational Linguistics.

[84] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer. Dialog act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26 (3):339–373, 2000.

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

[86] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling web-based acquisition of entailment relations. In D. Lin and D. Wu, editors, *Proceedings of EMNLP 2004*, pages 41–48, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[87] M. Tatu and D. Moldovan. A semantic approach to recognizing textual entailment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 371–378, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[88] M. Tatu and D. Moldovan. Cogex at rte 3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 22–27, Prague, June 2007. Association for Computational Linguistics.

[89] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967.

[90] R. Wang and G. Neumann. An accuracy-oriented divide-and-conquer strategy for recognizing textual entailment. In *Proceedings of the First Text Analysis Conference (TAC 2008)*, Gaithersburg, Maryland, USA, 2008.

[91] T. Wilson and J. Wiebe. Annotating attributions and private states. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 53–60, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[92] C. Zhang and J. Chai. What do we know about conversation participants: Experiments on conversation entailment. In *Proceedings of the SIGDIAL 2009 Conference*, pages 206–215, 2009.

[93] C. Zhang and J. Chai. An investigation of semantic representation in conversation entailment. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010.