Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Ask, and Shall You Receive? Understanding Desire Fulfillment in Natural Language Text

Snigdha Chaturvedi[1], Dan Goldwasser[2], Hal Daumé III[1]
[1]University of Maryland, College Park [2]Purdue University
snigdhac@cs.umd.edu, goldan55@gmail.com, hal@cs.umd.edu

Abstract

The ability to comprehend wishes or desires and their fulfillment is important to Natural Language Understanding. This paper introduces the task of identifying if a desire expressed by a subject in a given short piece of text was fulfilled. We propose various unstructured and structured models that capture fulfillment cues such as the subject's emotional state and actions. Our experiments with two different datasets demonstrate the importance of understanding the narrative and discourse structure to address this task.

1 Introduction

Understanding expressions of desire is a fundamental aspect of understanding intentional human-behavior. The strong connection between desires and the ability to plan and execute appropriate actions was studied extensively in contexts of rational agent behavior (Georgeff et al. 1999), and modeling human dialog interactions (Grosz and Sidner 1986).

In this paper we recognize the significant role that expressions of desire play in **natural language understanding**. Such expressions can be used to provide rationale for character behaviors when analyzing narrative text (Goyal, Riloff, and Daumé III 2010; Elson 2012a), extract information about human wishes (Goldberg et al. 2009), explain positive and negative sentiment in reviews, and support automatic curation of community forums by identifying unresolved issues raised by users.

We follow the intuition that at the heart of the applications mentioned above is the ability to recognize whether the expressed desire was fulfilled or not, and suggest a novel reading comprehension task: Given text, denoted as *Desire-expression* (e.g., "Before Lenin died, he said he wished to be buried beside his mother.") containing a desire ("be buried beside his mother") by the *Desire-subject* ("he"), and the subsequent text (denoted *Evidence fragments* or simply *Evidences*) appearing after the Desire-expression in the paragraph, we predict if the Desire-subject was successful in fulfilling their desire. Fig. 1 illustrates our setting.

Unlike most NLP tasks such as text categorization or sentiment classification which rely on lexical information, understanding desire fulfillment requires complex inferences connecting expression of desire, actions affecting the Desire-subject, and the extent to which these actions contribute to fulfilling the subject's goals. For example, in Fig. 1 the action of 'preserving' Lenin's body led to non-fulfillment of his desire.

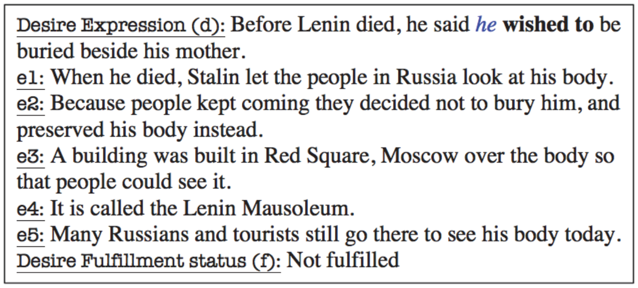

Desire Expression (d): Before Lenin died, he said *he* **wished to** be buried beside his mother.
e1: When he died, Stalin let the people in Russia look at his body.
e2: Because people kept coming they decided not to bury him, and preserved his body instead.
e3: A building was built in Red Square, Moscow over the body so that people could see it.
e4: It is called the Lenin Mausoleum.
e5: Many Russians and tourists still go there to see his body today.
Desire Fulfillment status (f): Not fulfilled

Figure 1: Example of a Desire Expression (d), Evidence fragments (e1. . .e5) and a binary Desire Fulfillment Status (f). The Desire-subject and Desire-verb are marked in blue and bold fonts respectively in the Desire-expression.

We address these complexities by representing the narrative flow of *Evidences*, and assessing if the events (and emotional states) mentioned in this flow contribute to (or provide indication of) fulfilling the desire expressed in the preceding Desire-expression. Following previous work on narrative representation, we track the events and states associated with the narrative's central character(s): the *Desire-subject* (Chambers and Jurafsky 2008; Chaturvedi et al. 2015; Srivastava, Chaturvedi, and Mitchell 2016).

While this representation captures important properties required by the desire-fulfillment prediction task, such as the actions taken by the Desire-subject, it does not provide us with an indication about the outcome of these actions. Recent attempts to support supervised learning of such detailed narrative structures by annotating data (Elson 2012b), result in highly complex structures even for restricted domains. Instead we model this information by associating a state, indicating if the outcome of an action (or the mention of an emotional state) provides evidence for making progress towards achieving the desired goal. We model the transitions between states as a latent sequence model, and use it to predict if the value of the final latent state in this sequence is indicative of a positive or negative prediction for our task.

We demonstrate the strength of our approach by comparing it against two strong baselines. First, we demonstrate the importance of analyzing the complete text by comparing with a textual-entailment based model that analyzes individual Evidence fragments independently. We then compare our latent structured model, which incorporates the narrative structure with an unstructured model, and show improvements in prediction performance. Our key contributions are:

- Introduce the problem of understanding desire fulfillment, and annotate and release two datasets.
- Present a latent structured model for this task, incorporating the narrative structure of the text, and propose relevant features that incorporate world knowledge.
- Empirically demonstrate that such a model outperforms competitive baselines.

1.1 Problem Setting

Our problem consists of instances of short texts (called *Desire-expressions*), which were collected in a manner so that each consists of an indication of a desire (characterized using a *Desire-verb*) by a *Desire-subject(s)*. The Desire-verb is identified by the following verb phrases: 'wanted to', 'wished to' or 'hoped to'.The three Desire-verbs were identified using lexical matches while the Desire-subject(s) was marked manually. Each *Desire-expression* is followed by five or fewer pieces of *Evidence fragment*s (or simply *Evidence*s). The Desire-expression and the Evidences (in order) consist of individual sentences that appeared contiguously in a paragraph. We address the binary classification task of predicting the *Desire Fulfillment* status, i.e. whether the indicated desire was fulfilled in the text, given the Evidences and the Desire-expression with Desire-verb and Subject identified. Fig. 1 shows an example of the problem.

2 Inference Models for Understanding Desire Fulfillment in Narrative Text

In this section we present three textual inference approaches, each following different assumptions when approaching the desire-fulfillment task, thus allowing a principled discussion about which aspects of the narrative text should be modeled.

Our first approach assumes the indication of desire fulfillment will be contained in a single Evidence fragment. We test this assumption by adapting the well-known Textual Entailment task to our settings, by generating entailment candidates from Desire-expression and Evidence fragments.

Our second approach assumes the decision depends on the Evidence text as a whole, rather than on a single Evidence fragment. We test this assumption by representing relevant information extracted from the entire Evidence text. This representation (depicted in Fig. 3) connects the central character in the narrative, the Desire-subject, with their actions and emotional states exhibited in the Evidence text. This representation is then used for feature extraction when training a binary classifier for the desire-fulfillment task.

Our final model provides a stronger structure for the actions and emotional states expressed in the Evidence text. The model treats individual Evidence fragments as parts of a plan carried out by the Desire-subject to achieve the desired goal, and makes judgments about the contribution of each step towards achieving the desired goal.

Data	Normalized?	P	R	F
MC Test	No	59.38	24.68	34.86
	Yes	76.09	45.45	**56.91**
Simple Wiki	No	50.00	2.22	4.26
	Yes	37.04	8.89	**14.34**

Table 1: Effects of normalizing the Desire-expression.

2.1 Textual Entailment (TE) Model

Recognizing Textual Entailment (RTE) is the task of recognizing the existence of an entailment relationship between two text fragments (Dagan et al. 2010). From this perspective, a textual entailment based method might be a natural way to address the desire fulfillment task. RTE systems often rely on aligning the entities appearing in the text fragments. Hence we reduce the desire fulfillment task into several RTE instances consisting of text-hypothesis pairs, by pairing the Desire-expression (hypothesis) with each of the Evidence fragments (text) in that example. However, we "normalized" the Desire-expression, so that it would be directly applicable for the RTE task. For example, the Desire-expression, "One day *Jerry* **wanted to** paint his barn.", gets converted to "Jerry painted his barn.". This process followed several steps:

- If the Desire-subject is pronominal, replace it with the appropriate named entity when possible (we used the Stanford CoreNLP coreference resolution system) (Manning et al. 2014).
- Ignore the content of the Desire-expression appearing before the Desire-subject.
- Remove the clause containing the Desire-verb ('wanted to', 'wished to' etc.), and convert the succeeding verb to its past tense.

The desire was considered 'fulfilled' if the RTE model predicted *entailment* for at least one of the text-hypothesis pairs of the example. E.g., the model could infer that the normalized Desire-expression example mentioned above, would be entailed by the following Evidence fragment- "It took Jerry six days to paint his barn that way." and hence it would conclude that the desire was fulfilled. Table 1 shows the performance of BIUTEE (Stern and Dagan 2012; Magnini et al. 2014), an RTE system, on the two datasets (see Sec. 4) used in our experiments[1]. Our results show that the RTE Model performs better with normalization. We use this model (with normalization) as a baseline in Sec. 5.

2.2 Unstructured Model

The Textual Entailment model described above assumes that the Desire-expression would be entailed by one of the individual Evidences. This assumption might not hold in all

[1] We also tested the TE model by using the default setting, optimized for the RTE task, however it performed very poorly.

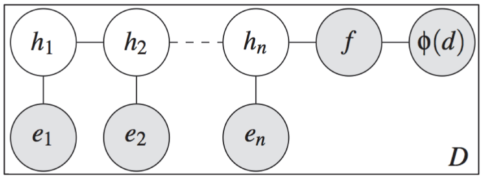


Figure 2: Structured model (LSNM) Diagram. Evidence e_i, Desire Fulfillment, f, and Structure-independent features, $\boldsymbol{\phi(d)}$, are observed. States, h_i, are hidden.

cases. Firstly, the indication of desire fulfillment (or its negation) can be subtle and expressed using indirect cues. More commonly, multiple Evidence fragments can collectively provide the cues needed to identify desire fulfillment. This suggests a need to treat the entire text as a whole when identifying cues about desire fulfillment.

We begin by identifying the Desire-subject and the desire expressed (using 'focal-word' described in Sec. 3) in the Desire-expression. Thereafter, we design several semantic features to model coreferent mentions of the Desire-subject, actions taken (and respective semantic-roles of the Desire-subject), and emotional state of the Desire-subject in the Evidences. We enhance this representation using several knowledge resources identifying word connotations (Feng et al. 2013) and relations. Fig. 3 presents a visual representation of this process and Sec. 3 presents further details.

Based on these features, extracted from the collection of all Evidences instead of individual Evidence fragments, we train supervised binary classifiers (*Unstructured models*).

2.3 Latent Structure Narrative Model (LSNM)

The Unstructured Model described above captures nuanced indications of desire-fulfillment, by associating the Desire-subject with actions, events and mental states. However, it ignores the narrative structure as it fails to model the 'flow of events' depicted in the transition between the Evidences.

Our principal hypothesis is that the input text presents a story. The events in the story describe the evolving attempts of the story's main character (the Desire-subject) to fulfill its desire. Therefore, it is essential to understand the flow of the story to make better judgments about its outcome.

We propose to model the evolution of the narrative using latent variables. We associate a latent state (denoted h_j), with each Evidence fragment (denoted e_j). The latent states take discrete values (out of H possible values, where H is a parameter to the model), which abstractly represent various degrees of optimism or pessimism with respect to fulfillment, f of the desire expressed in the Desire-expression, d. These latent states are arranged sequentially, in the order of occurrence of the corresponding Evidence fragments, and hence capture the evolution of the story (see Fig. 2).

The linear process assumed by our model can be summarized as: The model starts by predicting the latent state, h_0, based on the first Evidence, e_0. Thereafter, depending on the current latent state, and the content of the following Evidence fragment, the model transitions to another latent state. This process is repeated until all the Evidence fragments are associated with a latent state. We formulate the transition between narrative states as sequence prediction. We associate a set of *Content* features with each latent state, and *Evolution* features with the transitions between states.

Note that the desire fulfillment status, f, is viewed as an outcome of this inference process and is modeled as the last step of this chain using a discriminative classifier which makes its prediction based on the final latent state and a *Structure-independent* feature set, $\boldsymbol{\phi(d)}$. This feature set can be handcrafted to include information that could not be modeled by the latent states, such as long-range dependencies, and other cumulative features based on the Desire-expression, d, and the Evidence fragments, e_js.

We quantify these predictions using a linear model which depends on the various features, $\boldsymbol{\phi}$, and corresponding weights, $\mathbf{w}$. Using the Viterbi algorithm we can compute the score associated with the optimal state sequence, for a given input story as:

$$\text{score} = \max_{\mathbf{h}}[\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{e}, d, f, \mathbf{h})] \tag{1}$$

Algorithm 1 Training algorithm for LSNM

```
Input: Labeled set $\{(d, \mathbf{e}, f)_i \ \forall i \in \{1 \dots D\}\}$; and $T$: number of iterations
Output: Weights $\mathbf{w}$
Initialization: Initialize $\mathbf{w}$ randomly
for $t$ : 1 to $T$ do
    $\hat{\mathbf{h}}_i = \arg\max_{\mathbf{h}_i}[\mathbf{w_{t-1}} \cdot \boldsymbol{\phi}(\mathbf{e}_i, d_i, f, \mathbf{h}_i)]$ such that $f = f_i \forall i \in \{1 \dots D\}$
    $\mathbf{w_t}$ = StructuredPerceptron($\{(d, \mathbf{e}, \hat{\mathbf{h}}, f)_i\}$)
end for
```

Learning and Inference During training, we maximize the cumulative scores of all data instances using an iterative process (Alg. 1). Each iteration of this algorithm consists of two steps. In the first step, for every instance, it uses Viterbi algorithm (and weights from previous iteration, $\mathbf{w_{t-1}}$) to find the highest scoring latent state sequence, $\mathbf{h}$, that agrees with the provided label (the fulfillment state), f. In the following step, it uses the state sequence determined above to get refined weights for the t^{th} iteration, $\mathbf{w_t}$, using structured perceptron (Collins 2002). The algorithm is similar to an EM algorithm with 'hard' assignments albeit with a different objective. While testing, we use the learned weights and Viterbi decoding to compute the fulfillment state and the best scoring state sequence. Our approach is related to latent structured perceptron though we only use the last state (and structure-independent features) for prediction.

3 Features

We now describe our features and how they are used by the models. Table 2 defines our features and Fig. 3 describes their extraction for an example. They capture different semantic aspects of the desire-expression and evidences, such as entities, their actions and connotations, and their emotive states using lexical resources like Connotation Lexi-

Feature Type	Id	Definition
Entailment	F1	*TEPrediction*: Binary prediction of the Textual Entailment model (Stern and Dagan 2012).
Discourse	F2, F3	*ButPresent, SoPresent*: Binary features indicating if a ‘but’ or ‘so’ (respectively) followed the Desire-verb (‘wanted to’, ‘wished to’ etc.) in the Desire-expression.
Focal Word	F4, F5, F6	*focal count, focal syn and focal ant count:* Count of occurrences of the focal word(s), their WordNet (Miller 1995) synonyms and antonyms (respectively) in the Evidence. Occurrences of synonyms or antonyms were identified only when they had the same POS tag as the focal word(s).
	F7	*focal+syn count:* Sum of F4 and F5
	F8	*focal lemm count:* Count of occurrences of lemmatized forms of the focal word(s) in the Evidence.
Desire-sub. mention	F9	*sub count:* Count of all mentions (direct and co-referent) of the Desire-subject in the Evidence.
Emotional State	F10, F11	*+adj, -adj count:* Counts of occurrences of ‘positive’ and ‘negative’ adjectives (respectively) modifying the direct and co-referent mentions of the Desire-subject in the Evidence.
Action	F12, F13	*+Agent, -Agent count:* Number of times the connotation of verbs appearing in the Evidence agreed with and disagreed with (respectively) that of the intended action.
	F14, F15	*+Patient, -Patient count:* Count of occurrences of ‘positive’ and ‘negative’ verbs (respectively) in the Evidence which had the Desire-subject as the patient.
Sustenance	F16, F17	*isConforming, isDissenting:* Binary features indicating if the Evidence starts with a conforming or dissenting phrase (respectively). See Table 3 for example phrases.

Table 2: Feature definitions (Sec. 3). F1-F3 are extracted for each example while F4-F17 are extracted for each evidence.

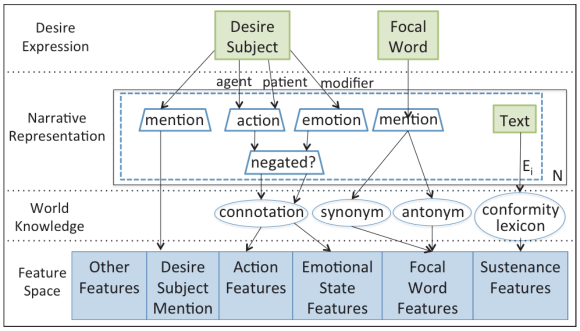


Figure 3: Framework for feature extraction for an example. E_i refers to the i^{th} evidence out of a total of N evidences.

Desire Expression: Amy wanted to **help** Ben, <u>so</u> she went with him.
e1: She also **offered** him a ride.
e2: Ben **thanked** her and **gifted** her a small present.
e3: Overall, she was *happy*.
Desire Fulfillment status: Positive (Fulfilled)

Figure 4: Artificial example indicating feature utility. The Desire-subject mentions are marked in blue, actions in bold and emotions in italics. Discourse feature is underlined.

con (Feng et al. 2013), WordNet and our lexicon of conforming and dissenting phrases. Before extracting features, we pre-processed the text [2] and extracted all adjectives and verbs (with their negation statuses and connotations) associated with the Desire-subject using dependency-parsing based rules.

1. Entailment (F1): This feature simply incorporates the *output of the Textual Entailment model.*

2. Discourse (F2-F3): These features aim to identify *indications of obstacles or progress* of desire fulfillment in the Desire-expression itself, based on discourse connectives. E.g. ‘so’ (underlined) in the Desire-expression in Fig. 4 indicates progress of desire fulfillment.

3. Focal words (F4-F8): These features identify the *word(s) most closely related* to the desire, and look for their presence in the Evidences. We define a *focal word* as the clausal complement of the Desire-verb (‘wanted to’, ‘hoped to’, ‘wished to’). If the clausal complement is a verb, the focal word is its past tense form. e.g., the focal word in the Desire expression in Fig. 4 is ‘helped’. A focal word is not simply the verb following the Desire-verb: e.g. in the Desire-expression in Fig. 1, the causal complement of ‘wished’ is ‘buried’. We then define features counting occurrences of the identified *focal words* and their WordNet *synonyms and antonyms* in each of the Evidences.

4. Desire-subject mentions (F9): This feature looks for mentions of Desire-subject in the Evidences assuming that a *lack of mentions* of the Subject might indicate absence of instances of their taking actions needed to fulfill the desire.

5. Emotional State (F10-F11): Signals about the fulfillment status could also emanate from the *emotional state of the Subject*. A happy or content Desire-subject can be indicative of a fulfilled desire (e.g. in Evidence e3 in Fig. 4), and vice versa. We quantify the emotional state of the Subject(s) using connotations of the adjectives modifying their mentions.

6. Action features (F12-F15): These features analyze the *intended* action and the actions taken by various entities. We first identify the *intended action* - the verb immediately following the Desire-verb in the Desire Expression. e.g., in Fig. 4 the intended action is to ‘help’. Thereafter, we design features that capture the connotative agreement between the intended action and the actions taken by the Desire-subject(s) in the Evidences. We also include features that describe connotations of actions (verbs) affecting the Desire-subject(s). E.g. in e1 of Fig. 4, the action by the Desire-subject (marked in blue), ‘offered’, is in connotative agreement with the intended action, ‘help’ (both have positive connotations according to Feng et al. (2013)). Also, the

[2] We obtained POS tags, dependency parses, and resolved co-references using Stanford CoreNLP (Manning et al. 2014).

Type	Phrases
Conforming	*in other words, for example, consequently*
Dissenting	*although, but, by contrast, conversely*

Table 3: Examples of conforming and dissenting phrases.

actions affecting the subject ('thanked', 'gifted') have positive connotations indicating desire fulfillment.

7. Sustenance Features (F16-F17): LSNM uses a chain of latent states to abstractly represent the content of the Evidences with respect to Desire fulfillment Status. At any point in the chain, the model has an expectation of the fulfillment status. The sustenance features *indicate if the expectation should intensify, remain the same or be reversed by the incoming Evidence fragment*. This is achieved by designing features indicating if the Evidence fragment starts with a 'conforming' or a 'dissenting' phrase. E.g. e3 in Fig. 4 starts with a conforming phrase, 'Overall', indicating that the fulfillment status expectation (positive in e2) should not change. Table 3 presents some examples of the two categories. These phrases were chosen using various discourse senses mentioned in Prasad et al. (2007). The complete list is available on the first author's webpage.

3.1 Unstructured Models

For the unstructured models, we directly used the Entailment and Discourse features (F1 to F3 in Table 2). For features F4 to F15, we summed their values across all Evidences of an instance. This ensured a constant size of the feature set in spite of variable number of Evidence fragments per instance.

3.2 Latent Structure Narrative Model

Our Structured model requires three types of features: (a) Content features that help the model assign latent states to Evidence fragments based on their content, (b) Evolution features that help in modeling the evolution of the story expressed by the Evidence fragments (c) Structure Independent features used while making the final prediction.

Content features: These features depend on the latent state of the model, h_j, and the content of the corresponding Evidence, e_j (expressed using features F4 to F15 in Table 2).

1. $\phi(h_j, e_j) = \alpha$ if the current state is h_j; 0 otherwise where $\alpha \in$ F4 to F15

Evolution features: These features depend on the current and previous latent states, h_j and h_{j-1} and/or the current Evidence fragment, e_j:

1. $\phi(h_{j-1}, h_j) = 1$ if previous state is h_{j-1} and current state is h_j; 0 otherwise.
2. $\phi(h_{j-1}, h_j, e_j) = \alpha$ if previous state is h_{j-1}, current state is h_j; 0 otherwise where $\alpha \in$ F16 and F17
3. $\phi(h_0) = 1$ if start state is h_0; 0 otherwise.

Structure Independent features $\boldsymbol{\phi(d)}$: This feature set is exactly same as that used by the Unstructured models.

4 Datasets

We have used two manually annotated datasets for our experiments: MCTest and SimpleWiki. Both the datasets are available on the first author's webpage.

Model	MCTest			SimpleWiki		
	P	**R**	**F**	**P**	**R**	**F**
BoW	41.2	50.0	45.2	28.2	20.0	23.4
TE	76.1	45.4	56.9	37.0	8.9	14.3
LR	70.6	63.2	66.7	50.0	8.9	15.2
DT	71.4	52.6	60.6	42.9	5.4	9.5
LSNM	69.6	84.2	**74.4**	37.5	21.3	**27.1**

Table 4: Test set performances. Our structured model, LSNM, outperforms the unstructured, TE and BoW models.

Collection and annotation: The MCTest data originated from the Machine Comprehension Test dataset (Richardson, Burges, and Renshaw 2013) which contained of a set of 660 stories and associated questions. The vocabulary and concepts are limited to the extent that the stories would be understandable by 7 year olds. We discard the questions and only consider the free text of the stories.

The SimpleWiki dataset was created from the textual content of an October, 2014 [3] dump of the Simple English Wikipedia (after discarding all lists, tables and titles). We chose Simple English Wikipedia over Wikipedia to limit the complexity of the vocabulary required to comprehend the content, thus making the task simpler and manageable.

The Desire-subject(s) and the Desire Fulfillment Status were manually annotated on CrowdFlower [4]. Each instance was annotated by 3 or more annotators as determined by CrowdFlower using expected annotation accuracy. Annotators were also required to demonstrate proficiency on an initial set of 5 test instances. To avoid annotator fatigue, each annotator was presented only 3 instances per session. The mean CrowdFlower confidence (inter-annotator agreement weighted by their trust scores) of the annotations was 0.92.

Training and Test Sets: The SimpleWiki and MCTest data consisted of about 1000 and 175 instances, 20% of which was held-out as test sets. In the test sets of SimpleWiki and MCTest, 28% and 56% of the data belonged to the positive (desire fulfilled) class respectively.

5 Empiricial Evaluation

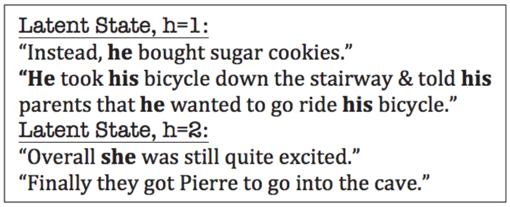
Table 4 compares the test set performances using F1 score of the positive (desire fulfilled) class for various models. We included a simple Logistic Regression baseline based on Bag-of-Words (BoW) features. For training the unstructured model, we experimented with different algorithms and show the results for the best two models: LR (Logistic Regression) and DT (Decision Trees). We report median performance values over 100 random restarts of our model since its performance depends on the initialization of the weights. Also, the number of latent states, H, was set to be 2 and 15 for the MCTest and SimpleWiki datasets respectively using cross-validation. The difference in optimal H values (and F1 scores) for the two datasets could be attributed to the difference in complexity of the language and concepts used in

[3] http://dumps.wikimedia.org/simplewiki/

[4] http://www.crowdflower.com/

Feature	F9	F13	F16	F17	F10
h=1	**0.682**	**0.502**	0.041	**0.024**	0.009
Overall	0.535	0.365	0.059	0.017	0.021
h=2	0.201	0.054	**0.098**	0.000	**0.049**

Table 5: Mean feature values for the overall dataset and the two latent states learned by LSNM on the MCTest dataset.


Latent State, h=1:

"Instead, **he** bought sugar cookies."

"He took **his** bicycle down the stairway & told **his** parents that **he** wanted to go ride **his** bicycle."

Latent State, h=2:

"Overall **she** was still quite excited."

"Finally they got Pierre to go into the cave."

Figure 5: Sample Evidences from the two latent states that our model learned on the MCTest data.

them. MCTest consists of children stories, focusing on simple concepts and goals (e.g., 'wanting to go skating') and their fulfillment is indicated explicitly, in simple and focused language (e.g., 'They went to the skating rink together.'). On the other hand, SimpleWiki describes real-life desires (e.g., 'wanting to conquer a country'), which require sophisticated planning over multiple steps, which may provide only indirect indication of the desire's fulfillment. This resulted in a harder classification problem, and increased the complexity of inference over several latent states.

The table shows that the unstructured models perform better than the TE model emphasizing the need for simultaneous analysis all of the Evidence text. Also, LSNM outperforms the unstructured models indicating the benefit of modeling the narrative structure of the Evidences.

5.1 Analysis of Latent States

We now delve into the contents of the latent states learned by LSNM and delineate their principal characteristics. Table 5 shows the mean values of some of the features for the two latent states learned by LSNM on the MCTest dataset. The table also shows the mean values for the overall dataset. We can see that Evidences assigned latent state=1 (h=1) were correlated with high values of features F9, F13 and F17, while those assigned h=2 had high values for features F16 and F10. This can also be observed in Fig. 5 which provides sample Evidences from the MCTest dataset assigned to the two latent states. We can see that the Evidences assigned *h=1* started with dissenting words like *Instead* (high value for F17) while those assigned h=2 started with consenting words like *overall, finally* (high value for F16). Also. in general Evidences with h=1 had greater number of mentions of the Desire-subject (marked in bold fonts) indicating high value for F9. Lastly, Evidences assigned h=2 also exhibit positive emotional state of the Desire-subject through words like *excited* (high value for F10).

6 Related Work

Expressions of desires and wishes have attracted psycholinguists (Shatz, Wellman, and Silber 1983) and linguists (Barak, Fazly, and Stevenson 2013) alike. Goldberg et al. (2009) detect wishes from text. Analyzing desires adds a new dimension to more general tasks like opinion mining (Pang and Lee 2007) where the manufacturers and advertisers want to discover users' desires or needs from online reviews etc. Another use-case would be in resolving issues for community forum users. For instance, the number of posts in Massive Open Online Courses forums often overwhelm the instructional staff (Chaturvedi, Goldwasser, and Daumé III 2014). Identifying posts containing unresolved issues can help focus the efforts of the instructional staff.

Our problem is related to Machine Comprehension (Richardson, Burges, and Renshaw 2013). However, unlike most systems, designed for understanding large textual collections (*macro-reading*) (Etzioni, Banko, and Cafarella 2006; Carlson et al. 2010), this work focuses on *Micro-reading*, understanding short pieces of text. Berant et al. (2014) also address micro-reading but with a different goal – answering domain-specific questions.

Our task is also related to Recognizing Textual Entailment (RTE) (Dagan et al. 2010; Dagan, Glickman, and Magnini 2006). However, we show that solving it additionally requires modeling the narrative structure of the text. There have been several attempts at modeling narrative structures which include narrative schemas (Chambers and Jurafsky 2009; 2008), plot units (Lehnert 1981) and Story Intention Graphs (Elson 2012b). Previous work has also studied connotations and word effects on narrative modeling (Feng et al. 2013; Goyal, Riloff, and Daumé III 2010). Our structured model and features, share similar motivation as these methods, while focusing on a specific classification task.

The AI task of recognizing plans of characters in a narrative viewing them as intentional agents (Mueller 2007; Wilensky 1978; Mani 2012) is also relevant. However, the focused nature of our task lets us employ latent variables to model the transitions between expectations and plans.

7 Conclusion

In this paper we have addressed the novel task of analyzing small pieces of text containing expression of a desire to identify if the desire was fulfilled in the given text. For solving this problem, we adopt three approaches based on different assumptions. We first use a textual entailment model to analyze small fragments of texts independently. Our second approach, an unstructured model, assumes that it is not sufficient to analyze different pieces of text independently. Instead, the complete text should be analyzed as a whole to identify desire fulfillment. Our third approach, a structured model, is based on the hypothesis that identifying desire fulfillment requires an understanding of the narrative structure and models the same using latent variables. We compare performances of these models on two different datasets that we have annotated and release. Our experiments establish the need to incorporate the narrative structure of the storyline offered by the text to better understand desire fulfillment.

References

Barak, L.; Fazly, A.; and Stevenson, S. 2013. Acquisition of desires before beliefs: A computational investigation. *Proceedings of CoNLL-2013*.

Berant, J.; Srikumar, V.; Chen, P.-C.; Vander Linden, A.; Harding, B.; Huang, B.; Clark, P.; and Manning, C. D. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1499–1510.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the 46th annual meeting of the Association for Computational Linguistics*.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, 602–610.

Chaturvedi, S.; Srivastava, S.; Daume, III, H.; and Dyer, C. 2015. Modeling Dynamic Relationships Between Characters in Literary Novels. *ArXiv e-prints, arXiv:1511.09376*.

Chaturvedi, S.; Goldwasser, D.; and Daumé III, H. 2014. Predicting instructor's intervention in mooc forums. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1501–1511. Baltimore, Maryland: Association for Computational Linguistics.

Collins, M. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1–8.

Dagan, I.; Dolan, B.; Magnini, B.; and Roth, D. 2010. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering* 16(01):105–105.

Dagan, I.; Glickman, O.; and Magnini, B. 2006. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges. Lecture Notes in Computer Science*, volume 3944, 177–190. Springer.

Elson, D. K. 2012a. Detecting story analogies from annotations of time, action and agency. In *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative*.

Elson, D. K. 2012b. Dramabank: Annotating agency in narrative discourse. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Etzioni, O.; Banko, M.; and Cafarella, M. J. 2006. Machine reading. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, 1517–1519.

Feng, S.; Kang, J. S.; Kuznetsova, P.; and Choi, Y. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Georgeff, M.; Pell, B.; Pollack, M.; Tambe, M.; and Wooldridge, M. 1999. The belief-desire-intention model of agency. In *Intelligent Agents V: Agents Theories, Architectures, and Languages*. Springer. 1–10.

Goldberg, A. B.; Fillmore, N.; Andrzejewski, D.; Xu, Z.; Gibson, B.; and Zhu, X. 2009. May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 263–271.

Goyal, A.; Riloff, E.; and Daumé III, H. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 77–86.

Grosz, B. J., and Sidner, C. L. 1986. Attention, intentions, and the structure of discourse. *Computational linguistics* 12(3):175–204.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

Magnini, B.; Zanoli, R.; Dagan, I.; Eichler, K.; Neumann, G.; Noh, T.; Padó, S.; Stern, A.; and Levy, O. 2014. The excitement open platform for textual inferences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, System Demonstrations*, 43–48.

Mani, I. 2012. *Computational Modeling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Mueller, E. T. 2007. Understanding goal-based stories through model finding and planning. *Magerko, B., and Riedl, M., eds., Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium* 95–101.

Pang, B., and Lee, L. 2007. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2(1-2):1–135.

Prasad, R.; Miltsakaki, E.; Dinesh, N.; Lee, A.; Joshi, A.; Robaldo, L.; and Webber, B. 2007. The penn discourse treebank 2.0 annotation manual. Technical report, University of Pennsylvania, Institute for Research in Cognitive Science.

Richardson, M.; Burges, C. J. C.; and Renshaw, E. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 193–203.

Shatz, M.; Wellman, H. M.; and Silber, S. 1983. The acquisition of mental verbs: A systematic investigation of the first reference to mental state. *Cognition* 14(3):301–321.

Srivastava, S.; Chaturvedi, S.; and Mitchell, T. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the Thirtieth AAAI Conference on Artifical Intelligence, AAAI 2016*.

Stern, A., and Dagan, I. 2012. BIUTEE: A modular open-source system for recognizing textual entailment. In *The 50th Annual Meeting of the Association for Computational Linguistics*, 73–78.

Wilensky, R. 1978. *Understanding Goal-based Stories.* Ph.D. Dissertation, New Haven, CT, USA. AAI7916531.