A STATISTICAL APPROACH FOR TEXT PROCESSING IN VIRTUAL HUMANS

Anton Leuski and David Traum
Institute for Creative Technologies
University of Southern California
Marina del Rey, CA, 90292, USA
{leuski,traum}@ict.usc.edu

ABSTRACT

We describe a text classification approach based on statistical language modeling. We show how this approach can be used for several natural language processing tasks in a virtual human system. Specifically, we show it can applied to language understanding, language generation, and character response selection tasks. We illustrate these applications with some experimental results.

1 INTRODUCTION

Interactive virtual characters has been shown to be effective tools in computer assisted training and simulation. They are already helping army personnel with developing negotiation, communication, and language skills. They teach officers how to structure an effective dialog, help them to learn to stay aware of foreign culture phenomena. One of the important tasks when creating a believable and engaging virtual character is making it capable of natural interaction with the users. Such a character should understand the user's speech and respond back appropriately. The virtual characters may play many different roles starting from delivering a single message to the user and supporting a limited dialog about the topic of the message (Leuski et al., 2006b) to very sophisticated virtual persona capable of engaging the user in a complex negotiation (Traum et al., 2005). The characters of different complexities share a common general design – the user's speech is converted from an audio signal into text by an automatic speech recognition (ASR); the text is analyzed by a language processing module that produces the character response; the text of the response is converted into sound by a text-to-speech (TTS) module and played back to the user. We focus on the natural language processing (NLP) part of this design. Generally a NLP module contains three components:

1. natural language understanding (NLU) module that interprets the text of the user's utterance and converts it into some internal representation;
2. dialog manager (DM) module that analyzes the interpretation and selects the appropriate response;
3. natural language generation (NLG) module that converts the internal representation to the text of the response.

In this paper we present a statistical text classification approach that is loosely based on a technique used in cross-lingual information retrieval. We show how this approach can be used to construct effective and robust NLU and NLG modules. Specifically, we describe its application in the negotiation training system called SASO[1] where a trainee interacts with a computer-controlled character (Traum et al., 2005).

We also show how this algorithm can be used to construct characters that do not require deep knowledge understanding and reasoning (Leuski et al., 2006b). The classification algorithm allows us to map directly from the text of the user's utterance to character responses.

2 VIRTUAL HUMAN SYSTEM

The Virtual Humans Project, at USC's Institute for Creative Technologies (ICT) and Information Sciences Institute (ISI), has the main goal of designing autonomous agents that support face-to-face interaction with people in many roles and in a variety of tasks. The agents must be embedded in the virtual world and perceive events in that world, as well as the actions of human participants. They must represent aspects of the dynamic situation in sufficient depth to plan

[1]SASO: Stability and Support Operations.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
DEC 2008	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Statistical Approach For Text Processing In Virtual Humans	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Institute for Creative Technologies University of Southern California Marina del Rey, CA, 90292, USA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002187. Proceedings of the Army Science Conference (26th) Held in Orlando, Florida on 1-4 December 2008

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**7**	

contingencies, develop beliefs and intentions, and form appropriate emotional reactions. They must communicate with each other and with human participants using multi-modal natural language communication.

One of our latest scenarios, the SASO system, includes a virtual human: a Spanish doctor. Set in a small Iraqi town plagued by violence, the human trainee takes on the role of an US Army captain with orders to move the doctor's clinic to a safer location.

The SASO training system architecture includes a large set of modules responsible for different components of the system (Hartholt et al., 2008). In the course of the interaction, the human trainee must negotiate with the virtual character, establishing trust and satisfying the objections of the doctor to moving the clinic. The system has a complex internal representation of the characters knowledge, it has a model of the domain, a model of itself and a model of the trainee. It incorporates a number of dialog strategies and is capable of reasoning about these concepts. The virtual human evaluates the utterances made by the trainee, updates its models of the conversational states and models of the trainee, and plans how to react and what to do next. The internal knowledge is represented in a form of semantic frames where each fact is encoded as a set of slot-value pairs. Figure 1 shows the NLP part of the system. We omit the rest of the system design for brevity. The user's speech is converted from an audio signal into text by an automatic speech recognition. The task of the NLU module is to convert a user's utterance into a semantic frame so the Dialog Manager can incorporate the user's input into its reasoning process. The reasoning and dialog management part of the system is represented in the figure by the part marked "Agent". The result of the process is yet another semantic frame that defines the character's response, e.g, a clarification question or a request for an action from the user. Then the task of the NLG module is to convert the frame into a text representation. The text is converted into sound by a text-to-speech module and played back to the user.

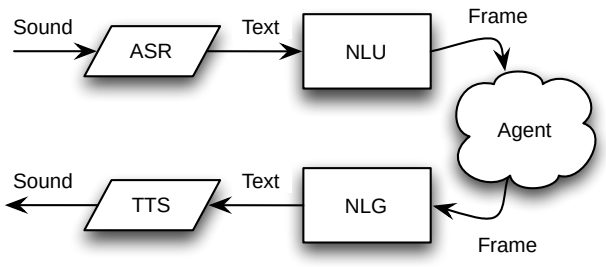

Figure 1: NLP part of a virtual human system

2.1 Natural Language Understanding

Figure 2 shows an example of a semantic frame used in the system corresponding to the utterance "I must move the clinic to the downtown area". The semantic frame is represented as a set of slot-value pairs and describes the speech act of the utterance and the individual parameters such agent, theme, etc. Each line in the Figure shows a slot name and the corresponding value separated by a white space. The task of the NLU module is to convert from the text of an utterance to the appropriate semantic frame.

```
mood declarative
sem.speechact.type statement
sem.modality.deontic must
sem.polarity positive
sem.type event
sem.event move
sem.agent captain-kirk
sem.theme clinic
sem.source market
sem.destination downtown
```

Figure 2: Semantic frame representing "I must move the clinic to the downtown area".

This task is known as the shallow semantic parsing task. With the availability of semantic resources like FrameNet (Baker et al., 1998) and more recently PropBank (Kingsbury, 2002) a number of statistical approaches have been developed in recent years that attempt to solve this problem (Gildea and Jurafsky, 2002; Pradhan et al., 2003; Fleischman et al., 2003). All these parsers use statistical training approaches with interesting combinations of features to obtain reasonable performances on general purpose semantic-parsing tasks. However, despite their good general purpose capabilities, these parsers have a number of problems when used in a virtual human system such as SASO. Firstly, they do not cover specic domains well, especially when these domains involve somewhat unusual terminology as it is the case of a detailed training simulation system. Secondly, while general purpose parsers tend to provide case-frame structures, that include the standard core case roles (Agent, Patient, Instrument, etc.), our system is dialogue oriented and it requires additional information about addressees, modality, speech acts, etc. These type of a semantic annotation does exist in publicly available resources. Thirdly, our system operates with transcriptions of spoken language, which is often different from the written language forms available for these general parsing approaches trained on collections of written English. Fourthly, the input for our parser comes from

an automatic speech recognition module that sometimes introduces errors into the transcription. These errors can affect the parsing performance significantly. Finally, our system operates in real time. Any noticeable delay in language processing decreases the system responsiveness and deteriorates the user's engagement. These general purpose parsing techniques rely on syntactic parsers (Charniak, 1997; Collins, 1997) that are not fast enough for real-time processing especially on long user utterances.

Our approach is different in that we view the shallow semantic parsing task as an information retrieval task. We assume that there is a fixed albeit sufficiently large number of significantly different inputs for the system's reasoning part and therefore a fixed number of important semantic frames. We assume that each frame is linked to a natural language utterance. These utterance-frame pairs are grouped per domain in separate framebanks, one for each character. Then the NLU task is given a ASR transcription of the user's utterance, find the most appropriate frame from the framebank. For example, this task is somewhat similar to web-based search, where an search system has to find web pages in response to short text query. In Section 3 we describe a statistical approach we used to implement our NLU module.

2.2 Natural Language Generation

The task of the NLG module is the direct opposite of the NLU task: given a frame constructed by the reasoning and dialog management module, find the appropriate natural language utterance. As with the NLU case we assume that we have a framebank of frame-utterance pairs. This NLG framebank is different from the NLU framebank because of some differences in the slot-value vocabulary between frames that the agent module accepts and the frames it generates.

2.3 Direct Language Mapping

The SASO doctor is a virtual human capable of complex interactions with a trainee. Often training simulations require less capable virtual human characters but in greater numbers. For example, we might need a character that is able to deliver a single message to the trainee and it is capable to answer a few questions on the topic of the message. We call these virtual human persona question-answering characters (Leuski et al., 2006a). Such a character does not need an extensive domain knowledge representation. We design these characters by directly mapping from the text of the trainee's utterance to the character response and allowing a text classification algorithm to learn the mapping from a sufficiently large set of examples. In the next section describe the statistical text processing technique that we use to solve the NLU, NLG, and direct language mapping tasks.

3 LANGUAGE MODELS

The key achievement of information retrieval is an ability to match two strings of text based on the content similarity. That is how a search system works – the text representation is computed for both documents and a query, a matching algorithm is applied, and the best match is returned to the person who entered the query. Different text content representation techniques exist. One of them is called statistical language modeling. A statistical language model is a probability distribution $P(W)$ over all possible word strings $W = w_1, ..., w_n$. A topic can be described by some amount of text sentences. The probability of observing a particular sentence describing the topic will vary from topic to topic. That way a language model can be used as a technique to represent the topic content. This concept is gaining a wide usage in information retrieval in recent years (Ponte and Croft, 1997).

Before we describe the details of the method, we have to make two observations. The first observation is that we can view a semantic frame as a sentence in a special language where slot-value pairs play the role of words. We can the define a language model over frames $P(F)$. That way we can compare a user's utterance to a semantic frame by computing the corresponding language models. Given a language model for a user's utterance we compare it to the language model of each known frame from the framebank and return the closest matching frame.

The second observation is that we cannot compare utterance and frame language models directly because they exist in different event spaces – the former is the probability over text utterances and the latter is the probability over frames. We can, however, compare a conditional probability of a frame given an observed text utterance $P(F|W)$ with the language models of known frames. One can interpret this value as a "translation" of the utterance W into the language of semantic frames. Here we use the framebank as the "parallel corpora" that maps strings in English to the corresponding strings in the frame language. The translation rules are implicitly derived from that mapping. This problem is similar to cross-language information retrieval task, e.g., where a search system has to find Chinese documents in response to an English

query (Grefenstette, 1998).

There are different ways to compare two probability distributions. In this paper we use the Kullback-Leibler (KL) divergence $D(P(F|W)||P(F))$ defined as

$$D(P(F|W)||P(F)) = \int_F P(F|W) \log \frac{P(F|W)}{P(F)} \quad (1)$$

which can be interpreted as the relative entropy between two distributions. Note that the Kullback-Leibler divergence is a dissimilarity measure, we use $-D(P(F|W)||P(F))$ to rank the frames.

Normally a topic is represented by a single text string. It is impossible to determine the language model from such a sample explicitly. The goal is to estimate the $P(W)$ as accurately as possible. The problem of estimating the joint probability $P(w_1, ..., w_n)$ of several words occurring together to form a string of text W has received a lot of attention in recent years among the researchers in the information retrieval community. The main challenge is to take into account the interdependencies that exist among the individual words while still making the computation feasible. Several different methods were suggested starting from the most trivial technique where all words assumed to be distributed identically and independently from each other – the unigram model – $P(w_1, ..., w_n) = \prod_{i=1}^{n} P(w_i)$. Other approaches include Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999) and Latent Dirichlet Allocation (LDA) (Blei et al., 2003), where the authors model text collections by a finite set of k topics and the joint probability is viewed as a mixture of the individual topic language models.

Lavrenko (Lavrenko, 2004) suggests a more general approach where the word interdependencies are defined by an unknown parameter vector θ and the words are taken as conditionally independent – they are independent for a given instance of the θ vector. It allows him to relax the independency assumption of the unigram model – the probability distribution does not depend on the oder of individual words, but it is affected by the co-occurences of individual words. With the help of the de Finetti's theorem, he showed that in this case the joint distribution can be represented as follows:

$$P(w_1, ..., w_n) = \int_{\theta \in \Theta} \left\{ \prod_{i=1}^{n} P_\theta(w_i) \right\} p(\theta)$$

The variable θ is the vector of hidden parameters, Θ is the set of all possible parameter settings. Each $P_\theta(w_i)$ is the appropriate probability distribution for individual words. The quantity $p(\theta)$ is a probability measure that tells us which parameter vector θ is a-priory more likely. The author gives several approximations for that expression for different Θ, $P_\cdot(w)$, and $p(\cdot)$. One this approximations is of a particular interest to us. It shows that given a set of training strings S, – e.g., all utterances from the framebank, – the joint distribution can be approximated as

$$P(w_1, ..., w_n) = \frac{1}{|S|} \sum_{s \in S} \prod_{i=1}^{m} p_s(w_i) \quad (2)$$

where $|S|$ is the size of the training set and $p_s(w_i)$ is the probability distribution of words in string s. There exist several estimations for the latter value. In this paper we use the Maximum Likelihood Estimation (MLE) with Jelinek-Mercer smoothing approach (Bahl et al., 1990):

$$\begin{aligned} p_s(w) &\cong \pi_s(w) \quad (3) \\ &= \lambda_\pi \frac{\#(w,s)}{|s|} + (1 - \lambda_\pi) \frac{\sum_s \#(w,s)}{\sum_s |s|} \end{aligned}$$

where $\#(w, s)$ is the number of times word w appears in string s, $|s|$ is the length of the string s, and the constant λ_π is the tunable parameter that can be determined from the training data.

Equation 2 assumes that all words w_i come from the same vocabulary. We can show that in the case of two different vocabularies, the joint distribution has the following form:

$$P(f, w_1, ..., w_n) = \frac{1}{|S|} \sum_{s \in \{F_s, W_s\}} \phi_{F_s}(f) \prod_{i=1}^{m} \pi_{W_s}(w_i)$$

Here s iterates over the set of training pairs that maps an utterance W_s to its frame interpretation F_s. $\phi_F(f)$ is the empirical probability distribution of slot-value pairs in frame F:

$$\phi_F(f) = \lambda_\phi \frac{\#(f,F)}{|F|} + (1 - \lambda_\phi) \frac{\sum \#(f,F)}{\sum |F|} \quad (4)$$

Combining these estimations we get the following expression for conditional probability of observing a particular slot-value pair f given a user's utterance W $P(f|W)$:

$$\begin{aligned} P(f|W) &= \frac{P(f, w_1, ..., w_n)}{P(w_1, ..., w_n)} \quad (5) \\ &= \frac{\sum_s \phi_{F_s}(f) \prod_{i=1}^{m} \pi_{W_s}(w_i)}{\sum_s \prod_{i=1}^{m} \pi_{W_s}(w_i)} \end{aligned}$$

The the matching criteria in Equation 1 can be written as

$$D(P(W)||P(F)) = \sum_f P(f|W) \log \frac{P(f|W)}{\phi_F(f)} \quad (6)$$

In summary, given a framebank $\{F_s, W_s\}$ and an utterance W, we use Equations 2, 3, 4, and 5 to compute Equation 6 for each frame F in the framebank and return the frame with the highest value $-D(P(W)||P(F))$.

Note one problem with this approach: the words in the utterance are assumed to be exchangeable, e.g., sentences "the area is secured" and "is the area secured" will have the same probabilities, which may potentially lead to questions interpreted as statements and vice versa. We deal with the problem by including local order dependencies in $\pi_s(w)$ in the form of a trigram model:

$$\begin{aligned} \pi_{3,s}(w) &= \lambda_1 \frac{\#(w_{-2}w_{-1}w, s)}{\#(w_{-2}w_{-1}, s)} \\ &+ \lambda_2 \frac{\sum_s \#(w_{-2}w_{-1}w, s)}{\sum_s \#(w_{-2}w_{-1}, s)} \\ &+ \lambda_3 \frac{\#(w_{-1}w, s)}{\#(w_{-1}, s)} + \lambda_4 \frac{\sum_s \#(w_{-1}w, s)}{\sum_s \#(w_{-1}, s)} \\ &+ \lambda_5 \frac{\#(w, s)}{|s|} + \lambda_6 \frac{\sum_s \#(w, s)}{\sum_s |s|} \end{aligned} \quad (7)$$

where $\sum_i \lambda_i = 1$. Here w_{-1} and w_{-2} are the words immediately preceding w in s.

This approach only returns frames that already exist in the framebank. The design guarantees that the output of NLU is always well-formed. At the same time we allow for significant amount uncertainty in the input for the modules, which makes the overall system flexible. However, it can also pose a problem if the set of frames in the framebank does not cover the domain of possible inputs. A previously unseen utterance can be misinterpreted and a wrong frame would be retrieved. One way to deal with this problem is to set a threshold on the KL divergence value. If the utterance similarity to the best matching frame is below that threshold, the NLU system returns what we call a "garbage" frame. It indicates to the agent that NLU failed to interpret its input. The agent might then ask the user to restate his question.

An alternative solution is to construct the frame instead of retrieving one. Equation 5 that defines the conditional probability of a particular slot-value pair f given an utterance W. We compute Equation 5 for every slot-value pair in the framebank and rank them by that number. We set a threshold on the probability value and return all slot-value pairs that are ranked above that threshold. This set makes the frame F corresponding to the utterance W. The threshold value can be determined by optimizing the algorithm on the train data. We call this approach *frame builder* in contrast to the former technique that we name *frame retriever*.

We use the same retrieval strategy for the NLG module. For this condition frames and utterances switch places. We rank all the utterances in the NLG framebank by the KL divergence $-D(P(F)||P(W))$, where the language model for the utterance $P(W)$ is estimated directly from the utterance text and the language model P(F) is computed by translating a frame into text using the NLG framebank:

$$D(P(F)||P(W)) = \sum_w P(w|F) \log \frac{P(w|F)}{\pi_W(w)} \quad (8)$$

where $P(w|F)$ is given by expression

$$P(w|F) = \frac{\sum_s \pi_{W_s}(w) \prod_{i=1}^m \phi_{F_s}(f_i)}{\sum_s \prod_{i=1}^m \phi_{F_s}(f_i)}$$

In case of a direct mapping between user's utterances and virtual human (VH) responses (see Section 2.3), we replace frame F in Equation 6 with a VH response English string $V = v_1...v_m$ and use the same technique to perform the selection. This approach might sound a bit counterintuitive: while in the previous two cases we translated from English to the language of semantic frames and vice versa, here we have English on both sides of the match. We could match a language model for a user utterance directly to the language model of the VH response and the translation (Equation 5) approach might seem redundant. We would argue that a user utterance and a VH response should still be viewed as drawn from different languages. A good example is a question-answer pair: "Who are you?" – "My name is Raed". In this example, both utterances touch upon the same topic – the name of the VH – but they are very unlikely to be expressed in the same words. The name of VH is much more likely to appear in the answer, than in the question. Some question specific grammar constructs (e.g., what, when, how, etc) and word sequences are much more likely to appear in the question than in the answer.

4 EXPERIMENTS

In our first set of experiments we evaluated the NLU module accuracy on the SASO framebank. The frame-

bank contains 1170 utterance-frame pairs with 51 unique frames. We randomly split that set into 1053 training and 117 testing examples. We experimented with both frame retriever and frame builder system variants. Table 1 shows the slot-value pair F-score for each system in percentages. F-score is geometric mean of recall and precision. Recall is the percentage of slot-value pairs that were returned by the system to the number of pairs that must be present in the NLU output. Precision is the percentage of correct slot-value pairs among all slot-value pairs returned by the system.

We considered two experimental conditions: 1) trained on hand transcribed text and tested on transcribed text and 2) trained on transcribed text and tested on the results from the automatic speech recognition. The table shows that frame retriever generally outperforms the frame builder system. Our analysis shows that

Another factor we considered was the quality of the NLU input. We compared its performance on the transcribed data versus its performance on the ASR output. We observed 8.5-9% drop in accuracy when evaluating on ASR output. The average word error score (WER) between the transcribed test utterances and the ASR output was 35.6%.

We also evaluated the frame retriever system where the unigram estimation for the probability of words in a string (Equation 3) was replaced with the trigram estimation (Equation 7). The goal was to capture short distance dependencies between words, e.g., allowing the algorithm to distinguish between questions and statements. We observed 1.5% improvement on the performance. That improvement was not statistically significant.

Data set	builder	retriever
Trans	81.45	84.48
ASR	74.46	76.81

Table 1: Classification accuracy for the different NLU approaches. The accuracy numbers are given in percentages.

For the NLG experiments we used 220 frame-utterance pairs. We split the set randomly into 198 training and 22 test samples. We trained the algorithm on the 198 training examples, and used it to generate a single (highest-ranked) utterance for each example in both the test and training sets. The success rate was 96% for training examples and 90% for test examples (DeVault et al., 2008).

	# of questions	# of answers	SVM	LM	
					impr.
1	238	22	44.12	47.90	8.57*
2	120	15	63.33	64.17	1.32
3	150	23	42.67	50.00	17.19*
4	108	18	42.59	50.00	17.39*
5	149	33	32.21	42.86	33.04*
6	39	8	69.23	66.67	-3.70
7	135	31	42.96	50.39	17.28*
8	1261	60	53.13	61.99	16.67

Table 2: Comparison of two different algorithms for answer selection on 8 QA characters. The table shows the number of answers and the number of questions collected for each character. The accuracy and the improvement over the baseline numbers are given in percentages.

For our last set of experiments we evaluated the direct language mapping approach for building question-answering characters. Table 2 shows the results of this comparison for eight characters created for different ICT projects. Each character has collection of responses and sample questions linked to those responses. We divided each collection of questions into training and testing subsets and evaluated the system following the 10-fold cross-validation schema. We compared the accuracy of the responses of our technique to the performance of a state-of-the-art text classification system that uses Support Vector Machine approach.

We observe that the language model approach is more successful for problems with more answer classes and more training data. The table shows the percent improvement in classication accuracy for the LM-based approach over the SVM baseline. The asterisks indicate statistical signicance using a t-test with the cutoff set to 5% ($p < 0.05$). More details about this experiment can be found elsewhere (Leuski et al., 2006b).

5 CONCLUSIONS

In this paper we describe a text classification approach based on statistical language modeling. We showed how the approach can be used in a virtual human simulation system language processing pipeline. We showed how we can build both language understanding and language generation components of the system using this algorithm. We also showed an implementation where the classification algorithm takes the role of the whole language pipeline mapping directly from the trainee's utterances to the character responses.

ACKNOWLEDGMENTS

The authors would like to thank David DeVault, Ron Artstein, and Rahul Bhagat for their help.

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Bahl, L. R., Jelinek, F., and Mercer, R. L. 1990. A maximum likelihood approach to continuous speech recognition. In *Readings in speech recognition*, pages 308–319. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

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

Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Charniak, E. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings AAAI/IAAI*, pages 598–603. AAAI Press/MIT Press.

Collins, M. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23.

DeVault, D., Traum, D., and Artstein, R. 2008. Making grammar-based generation easier to deploy in dialogue systems. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 198–207, Columbus, Ohio, June. Association for Computational Linguistics.

Fleischman, M., Kwon, N., and Hovy, E. 2003. Maximum entropy models for framenet classification. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 49–56, Morristown, NJ, USA. Association for Computational Linguistics.

Gildea, D. and Jurafsky, D. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3).

Grefenstette, G. 1998. *Cross-Language Information Retrieval*. Kluwer Academic Publishers, Norwell, MA, USA.

Hartholt, A., Russ, T., Traum, D., Hovy, E., and Robinson, S. 2008. A common ground for virtual humans: Using an ontology in a natural language oriented virtual human architecture. In *Proceedings of Language Resources and Evaluation Conference (LREC)*.

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International ACM SIGIR Conference*, pages 50–57.

Kingsbury, P. 2002. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology Conference*.

Lavrenko, V. 2004. *A Generative Theory of Relevance*. Ph.D. thesis, University of Massachusetts at Amherst.

Leuski, A., Pair, J., Traum, D., McNerney, P. J., Georgiou, P., and Patel, R. 2006a. How to talk to a hologram. In *Proceedings of the 11th international conference on Intelligent user interfaces (IUI'06)*, pages 360–362, Sydney, Australia, January. ACM Press New York, NY, USA.

Leuski, A., Patel, R., Traum, D., and Kennedy, B. 2006b. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, Sydney, Australia, July.

Ponte, J. M. and Croft, W. B. 1997. Text segmentation by topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 120–129.

Pradhan, S., Hacioglu, K., Ward, W., Martin, J., and Jurafsky, D. 2003. Semantic role parsing: adding semantic structure to unstructured text. *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 629–632, Nov.

Traum, D., Swartout, W., Gratch, J., Marsella, S., Kenney, P., Hovy, E., Narayanan, S., Fast, E., Martinovski, B., Bhagat, R., Robinson, S., Marshall, A., Wang, D., Gandhe, S., and Leuski, A. 2005. Dealing with doctors: Virtual humans for non-team interaction training. In *Proceedings of ACL/ISCA 6th SIGdial Workshop on Discourse and Dialogue*, Lisbon, Portugal, September.