A Conversational Movie Search System Based on Conditional Random Fields

Jingjing Liu, Scott Cyphers, Panupong Pasupat, Ian McGraw, Jim Glass

MIT Computer Science & Artificial Intelligence Laboratory, Cambridge, MA 02139, U.S.A.

{jingl, cyphers, ppasupat, imcgraw, glass}@csail.mit.edu

Abstract

Online streaming companies such as Netflix have become dominant in the media distribution sector. However, such media delivery services often support very rudimentary search, especially for natural language queries. To provide a more natural search interface, we have developed a conversational movie search system, which parses the recognition hypothesis of a spoken query into semantic classes using conditional random fields (CRFs), and then searches an indexed database with the identified semantics. Topic modeling on user-generated content (e.g., movie reviews) is employed for query expansion. Thirteen searching schemas are supported (such as genre, plot, character and soundtrack search). A crowd-sourcing platform was utilized to automatically collect large-scale annotated data for incremental CRF training.

Index Terms: conditional random fields, spoken dialogue system

1. Introduction

With the rapid increase of speed of computation and network bandwidth, online streaming has become increasingly dominant in the media distribution sector, including services such as Netflix, Amazon Video on Demand, and Apple iTunes. However, the focus of these services is often on non-latency media delivery, while search support is often basic, especially for natural language queries. For example, if one types in the query "show me a funny movie about bridesmaids starring Kera Nightley" into Netflix, the search results will list two irrelevant actors ("Jeff Bridges" and "Udo Kier") as the top two hits. This is likely due to two main challenges: 1) semantic interpretation of queries; and 2) search over unstructured content.

To interpret and handle a search query semantically, a more advanced language understanding mechanism is required, which could parse the query "car chase crime movies in the 1990s" into several semantic slots ("car chase," "crime movies," and "in the 1990s") and identify the semantic meaning of each slot ("car chase: PLOT," "crime: GENRE," "1990s: YEAR"). A more advanced database search strategy is also required, in order to search an indexed database on these various semantic slots spontaneously and retrieve the union/intersection of relevant hits.

Semantic language understanding has been a major challenge in dialogue systems. Many systems have employed context-free grammar (CFG) for sentence parsing and semantic understanding [1][2]. Because CFG-based language understanding is expert-controlled, it maintains high precision on closed-set utterances. However, the coverage of a grammar is limited, and the parsing is poor for out-of-vocabulary words and misspellings, as well as unseen linguistic patterns. For example, given the query "show me a funny movie about bridesmaids starring Kera Nightley," it is hard for a CFG to identify the misspelled "Kera Nightley" as an actor. It is also impractical to cover all possible plot keywords with a closed-set grammar, such as "bridesmaids," "werewolves" and "wizards." Developing a CFG requires a lot of expert knowledge and human effort; thus is not easy to scale to larger datasets or generalize to other domains.

To bring in more flexibility and less expert involvement, a data-driven approach is a plausible alternative. There have been many studies on language understanding using statistical methods such as CRF [3] or semi-CRF [4][5]. In this work, we explore the challenges in applying sequential labeling approaches to real dialogue systems for spoken language understanding. For a prototype demonstration, we develop a conversational movie search system, which employs semi-CRFs to handle natural language queries. As shown in Figure 1, given the query from a user, a CRF parser would segment the query constituents and assign a semantic class to each segment (GENRE: funny; PLOT: bridesmaids; ACTOR: Kera Nightley).

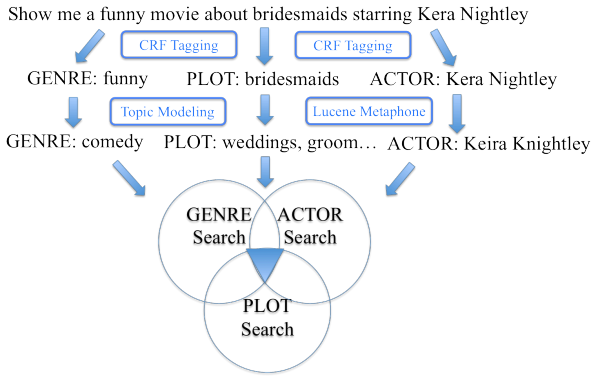

Figure 1: *Example of advanced query interpretation and database search process. The upper layer of query semantic tagging identifies the semantic class of each query segment, and the bottom layer of multi-field database search retrieves relevant hits conjunctionally.*

These labeled query terms could be used to search a relational indexed database. However, using exact match for search may have data sparsity problems. Take the IMDB movie database as an example. There are only 20 genres listed (e.g., "Comedy" "Thriller" "Romance"), and "funny" is not included. Also, in some movie plots, there might not be the exact query word ("bridesmaids"), but instead, other relevant words may occur (e.g., "weddings," "groom"). To expand query terms to such synonyms, we utilize topic clustering for semantic query generalization. The most representative topic clusters can be learned from movie documents (e.g., plots and reviews) using topic models such as Latent Dirichlet Allocation (LDA) [6]. As shown in the second step of Figure 1, the identified query terms will be generalized to relevant topics (GENRE: funny → comedy; PLOT: bridesmaids → weddings, groom).

Topic models do not handle misspellings though. To normalize misspelled query terms, we utilize the Lucene[1] search engine for advanced database retrieval. For example, Lucene supports a "metaphone" search, which converts query terms to standard phonemes and searches the database by pronunciations instead of spellings (as shown in Figure 1). The normalized query constituents are then subjected to the multi-field search engine to retrieve the conjunction/union of relevant hits.

By integrating the CRF-based semantic tagging, topic modeling based query expansion, and Lucene-based search into a spoken dialogue framework, we have constructed a conversational movie search system, *MovieBrowser*, which allows users to search movies by genres, actors, quotes, soundtracks, etc. Table 1 shows a real dialogue between a user and the prototype system.

Table 1. *A dialogue between a user ("U") and our prototype system MovieBrowser ("M").*

U: Find me a must-see sci-fi movie about aliens from the last five years.
M: I found 3 highly rated sci-fi movies about aliens since 2007: Avatar, District 9, and Star Trek.
U: Find me the movie with the song "I Feel Pretty".
M: West Side Story was a crime and drama movie released in 1961. It had the song "I Feel Pretty".
U: I want to see an adventure movie about werewolves and vampires.
M: Here are 2 adventure movies about werewolves and vampire: The Twilight Saga: Eclipse and Underworld: Rise of the Lycans.
U: Show me the pirate movie with "Jack Sparrow".
M: Here are 4 Jack Sparrow movies about pirate: The Curse of the Black Pearl, Dean Man's Chest, On Stranger Tides, and At World's End. Which one would you like to watch?
U: Play the trailer of "On Stranger Tides".
M: (Play the trailer)

2. Conditional Random Fields

This section gives a background of Conditional Random Fields [3]. CRFs model the conditional probability of a label sequence given the input. For example, given an input word sequence $x = (x_1, x_2, \ldots, x_M)$, the goal is to find a label sequence $y = (y_1, y_2, \ldots, y_M)$, which maximizes:

$$p(y|x) = \frac{1}{Z_\lambda(x)} \exp\left\{\sum_{i=1}^{M+1} \lambda \cdot f(y_{i-1}, y_i, x, i)\right\} \quad (1)$$

where the partition function $Z_\lambda(x)$ is a normalization factor, and λ is a weight vector.

For segment-based semantic tagging, given the word sequence $x = (x_1, x_2, \ldots, x_M)$, the goal is to find $s = (s_1, s_2, \ldots, s_N)$, which denotes a segmentation of the input as well as a classification of all segments. Each segment is represented by a tuple $s_j = (u_j, v_j, y_j)$. Here u_j and v_j are the start and end indices of the segment, and y_j is a class label. Semi-Markov CRFs can be used to model the segmentation and classification problem jointly:

$$p(s|x) = \frac{1}{Z_\lambda(x)} \exp\left\{\sum_{j=1}^{N+1} \lambda \cdot f(s_{j-1}, s_j, x)\right\} \quad (2)$$

where $f(s_{j-1}, s_j, x)$ is a vector of feature functions defined on segments. For example, a segment-based lexical feature is given by:

$$f(s_{j-1}, s_j, x) = \delta(s_j \epsilon L)\delta(y_j = b) \quad (3)$$

where L denotes a lexicon, b denotes a class, and $\delta(s_j \epsilon L)$ denotes that the current segment matches an element in lexicon L. More precisely, f is of the function form $f(y_{j-1}, y_j, x, u_j, v_j)$. Given labeled sentences, we estimate λ in (2) that maximizes the conditional likelihood of training data while regularizing model parameters. The learned model is then used to predict the label sequence s for a future input sequence x.

3. MovieBrowser System

In this project, we substantiate the CRF-based semantic tagging approach in a spoken dialogue system, *MovieBrowser*, which is a web-based multimodal movie search engine. Figure 2 shows the architecture of the system, which contains three major parts: 1) CRF-based language understanding; 2) speech and dialogue; and 3) advanced database search.

When a user submits a spoken query via the web-based interface of the system, trained CRF models will parse the recognition hypothesis of the spoken utterance. The segmented/labeled query is then sent to the search engine to search on the indexed database, which supports various search schemas, such as plot search, character search and review search. The retrieved results are sent to the dialogue management and language generation components to generate both graphical and spoken responses, which are sent back to the user via the web-based interface.

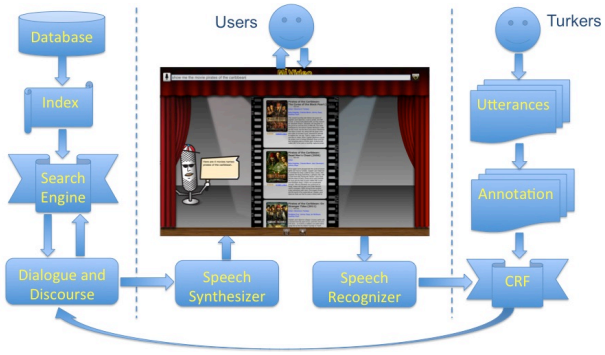

Figure 2: *System architecture of MovieBrowser* system.

3.1. Language Understanding

CRFs are discriminative graphical models. To collect large-scale training data, we employ crowd-sourcing platforms (e.g., Amazon Mechanical Turk[2]) to collect annotated utterances from hired workers (or turkers). Turkers make up query sentences based on instructions and scenarios provided, and other turkers label each sentence with pre-provided classes. CRF models are trained on these annotated data and embedded in the dialogue system for spoken utterance parsing.

[1] http://lucene.apache.org/core/

[2] https://www.mturk.com/mturk/welcome

The features used for model training include word features (*n*-grams in training data), lexical features (a segment of the sentence matching a word/phrase in a lexicon), regular expression features, and transition features. More details of model training and features can be found in [4][5].

3.2. Database Search

We collected a movie database involving over 2 million movies via IMDB API[1]. To build the prototype system, we selected 12k most popular movies as an initial set. We also collected over 8,000 critics' reviews on these selected movies via the Rotten Tomatoes API[2].

We then built a multi-field Lucene index for the movie database. The major fields in the index include title, director, actors, plot summaries, etc. A user's query will be parsed into these fields by CRF labeling, and a conjunction query will be generated to search the database. Various Lucene search algorithms are utilized, such as fuzzy match for titles and plots, and metaphone match (pronunciation search) for actors and directors. For example, as exemplified in Figure 1, "Kera Nightley" can be generalized to "Metaphone: KR NTL" by the Lucene query parser, and the actor "Keira Knightley" will be retrieved through the "metaphone" match process.

3.3. Topic Modeling

Database search often suffers from data sparsity problems. For example, there are only 20 genres in the IMDB database (e.g., "comedy," "thriller"). But real users might ask for "funny movies" or "scary films." To capture these synonyms for query expansion, we employ topic models as a pre-process.

As aforementioned, we collected a database with over 2 million movie plots as well as ~8000 movie reviews. To learn the most representative topics from these documents, we apply LDA topic clustering to the two data sets. Table 2 and Table 3 show some clustering examples on each set.

Table 2. *Examples of topic clusters on movie reviews.*

Cluster	Extended topics
action	fun entertaining entertainment high fast adventure flick hard
comedy	funny humor comic laugh joke fun satire amusing wit hilarious
history	war life documentary portrait political compelling human truth
family	kids children sweet disney animation charming tale musical
romance	love romantic women drama emotional girl boy life fine
thriller	horror dead suspense scary violence blood psychological gore

Table 3. *Examples of topic clusters on movie plot summaries.*

Cluster	Extended topics
detective	police murder killer investigation suspect evidence victim
religion	god church priest catholic christ holy heaven cult bishop bible
war	army soldier military battle nazi russian officer vietnam enemy
crime	thief robbery kidnapped hostage ransom mob gangster mafia
sports	team game football coach player season league baseball
music	band rock singer song concert record hip musician tour pop
spirit	ghost evil haunted devil hell demon demons supernatural curse
aliens	earth planet space alien human moon destroy mission universe
fairytale	princess castle kingdom emperor palace knight duke throne
violence	death fear tragedy violent struggle confront guilt abuse
magic	evil monster witch dragon fairy magician monsters spell wizard

[1] http://www.imdb.com/interfaces
[2] http://developer.rottentomatoes.com/

After CRF labeling, the parsed utterance is subjected to a query expansion process, which replaces the labeled genres to the more general classes (e.g., mapping "funny" to "comedy"), as well as including the topics within the same cluster of a plot keyword as extended query terms (e.g., "army," "soldier," and "battle" for "war"). These learned topics are also used as vocabularies for recognizer training.

3.4. Speech and Dialogue

For the speech recognition of users' utterances, we use the SUMMIT system [7], the acoustic models of which are trained with an English corpus unrelated to this domain. The class *n*-gram language model is trained by parsing the same corpus used for CRF training. To make the interaction between users and the system more natural, we implement a preliminary dialogue and discourse framework to support spoken conversations, as exemplified in Table 1. Since the dialogue in the movie domain is very straightforward, we deploy a set of heuristic rules for context resolution.

Figure 3 shows a screenshot of the system. The WAMI toolkit [8] was used to integrate speech and natural language processing components into a Web-based interface. The system accepts both spoken and typed queries. Spoken responses from the system summarize the search results. Retrieved movies show up on the screen with detailed information such as the cast, plot summary and critics' reviews, ranked by the popularity of the movies.

Figure 3: *Screenshot of the MovieBrowser system.*

4. Experiments and Evaluation

In this section, we present a systematic evaluation of the proposed approaches conducted on real data. For the data collection, three Amazon Mechanical Turk (AMT) tasks were conducted sequentially. The first one was for learning the scope of users' queries. In this task, turkers were asked to type in any inquiries about movies. 500 sentences were collected from this task, most of which fall into the following categories: titles, actors, mpaa ratings, viewers' ratings, reviews, directors, characters, soundtracks, trailers, release dates, genres, quotes, and plot synopsis.

Based on these sampled sentences, we defined a schema of 13 semantic classes and collected training data annotated with these class labels. For quality control of the annotations, we conducted a reversed AMT task by giving turkers the annotated results and asking them to make up original sentences. As shown in Table 4, we provided a list of e-forms with pairs of

<CLASS>:<value>. Given each e-form, a turker was asked to create a natural language query to search for the movies specified by the information provided.

Table 4. *Examples of e-forms for movie query collection.*

ACTOR: Judith Dench; GENRE: film noir; YEAR: last decade
GENRE: action; PLOT: prisoners; VIEWERS' RATINGS: highly rated
PLOT: motor racing; DIRECTOR: John Rebel; YEAR: past seven years
GENRE: children; MPAA RATING: PG13; PLOT: sibling rivalry

The category values (e.g., movie titles, actors' names) in the e-forms were divided into two separate sets for the training and test data. A total of 4,384 sentences were collected through this task, to form "Dataset I." The training set contains 2,175 sentences and the test set contains 2,209 sentences.

To collect sentences with free values, we set up a third AMT task, where turkers were asked to make up movie query sentences based on these pre-defined classes. The collected sentences were then subjected to another annotation task, where other turkers labeled the sentences with the provided classes. Figure 4 shows the interface of the AMT annotation task. Given a sentence, a turker could select any segment of the sentence and assign one of the displayed classes. Multiple segments could be labeled for each sentence. A total of 5200 sentences were collected and annotated through this task. We named this as "Dataset II" and randomly divided it into training and test sets.

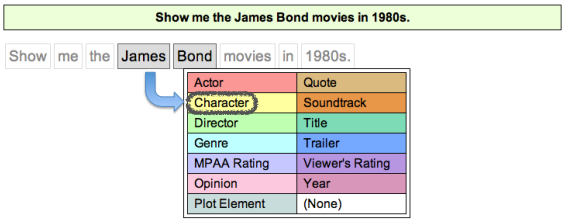

Figure 4: *Screenshot of the annotation task on AMT. The turker selected "James Bond," and a dropdown list with 13 classes showed up. The turker could select one of the labels (e.g., "Character") for the current segment.*

Table 5. *Experimental results of CRF tagging on two datasets.*

	Recall		Precision		F1	
Test set	**I**	**II**	**I**	**II**	**I**	**II**
CRF	54.35	58.01	62.31	62.12	58.05	60.00
Semi-CRF	63.24	63.37	68.27	68.82	65.66	65.98
Lexical features	83.18	65.79	76.47	71.93	79.68	68.72

We utilized an open-source package[1] for implementing CRF. For lexical features, we used a lexicon of ~12k movie titles, ~200k actors' names, and ~6k directors' names from IMDB database. Table 5 shows the tagging results on Datasets I and II respectively. Semi-CRF outperforms CRF on both datasets, and using lexical features helps improve the performance. The recall on Dataset I is significantly higher than that on Dataset II when using lexical features. This is understandable as the sentences in Dataset I were collected based on provided e-forms with formal lexicons (e.g., official movie titles and actors' full names). Thus these lexicon words are easier to capture by lexical features than the free-value sentences in Dataset II.

[1] http://crf.sourceforge.net/

To evaluate the database search performance, we put the dialogue system on Amazon Mechanical Turk and conducted a user experience task, where turkers interact with the real system. To avoid the interference of recognition errors, we collected typed-in queries. The evaluation on speech recognition can be found in [9]. A total of 1000 search events were collected via this task. In each search event, a turker submitted a query, and the system retrieved relevant movies from the database and showed the top-10 ranked on the screen. The turker could select a checkbox on each movie if he/she thinks the movie is relevant to his/her query. Among these 1000 search events, the average rank of the top movie that was identified as relevant is 0.65 (between 0 and 9). This shows that the top or the second-top hit typically satisfies the user's query. The collected utterances are used to retrain the CRFs incrementally on the fly [9].

5. Conclusions

This paper has presented a conversational movie search system, which parses the recognition hypothesis of a spoken query into semantic class labels using conditional random fields (CRFs), and searches an indexed movie database with the identified semantics for multi-field retrieval. Topic models are employed for query expansion and vocabulary learning, and various searching schemas are supported for advanced database search. We also utilized a crowd-sourcing platform to automatically collect large-scale annotated data for CRF training and evaluation.

For future work, we will explore more advanced dialogue and discourse mechanisms to enhance the dialogue capability of the system. More free-value training data will be collected from real users via crowd sourcing to incrementally improve CRF models and recognition.

6. Acknowledgements

This research is supported by Quanta Computers, Inc. through the T-Party project, and by Google. Thanks to Willie Walker for leading the user interface development. Also thanks to Victor Zue and Stephanie Seneff for helpful discussions.

7. References

[1] A. Gruenstein and S. Seneff. Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms. In Proc. of SIGDIAL 2007.

[2] J. Liu and S. Seneff. A Dialogue System for Accessing Drug Reviews. In Proc. of ASRU 2011.

[3] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proc. of ICML, 2001.

[4] X. Li. Understanding the Semantic Structure of Noun Phrase Queries. In Proc. of ACL, 2010.

[5] J. Liu, X. Li, A. Acero, and Y-Y Wang. Lexicon Modeling for Query Understanding. In Proc. of ICASSP 2011.

[6] D. M. Blei, A. Y. Ng, M. Jordan. J. Lafferty. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (4–5). 2003.

[7] J. Glass. A Probabilistic Framework for Segment-Based Speech Recognition. Computer Speech and Language 17, 137-152, 2003.

[8] A. Gruenstein, I. McGraw, and I. Badr. The WAMI Toolkit for Developing, Deploying, and Evaluating Web-Accessible Multimodal Interfaces. In Proc. of ICMI, 2008.

[9] I. McGraw, S. Cyphers, P. Pasupat, J. Liu, J. Glass. Automating Crowd-Supervised Learning for Spoken Language Systems. In Proc. of Interspeech 2012.