December 1999

A Natural Language Interface for Information Retrieval from Forms on the World Wide Web

Frank Meng
University of California, Los Angeles

A NATURAL LANGUAGE INTERFACE FOR INFORMATION RETRIEVAL FROM FORMS ON THE WORLD WIDE WEB

Frank Meng
University of California, Los Angeles
U.S.A.

Abstract

This paper presents an approach for retrieving information from forms on the world wide web from natural language input. The structured nature of the form can be utilized to process natural language input for querying data sources on the web that provide form interfaces. Since the valid values for each field can be determined from the form itself or by a user of the form, the form can be filled out be looking for these values in the natural language user input. Since it is possible for a particular value to be valid for more than one field, the surrounding context must be used to determine the correct field for an ambiguous value. A statistical disambiguation method based on n-gram statistics is proposed. It was shown that this method works better than using single context words for disambiguation when the domain is limited.

1. INTRODUCTION

Much data on the world wide web is made available through a form interface, where the user supplies values for fields and information is retrieved reflecting how the form was filled out. The advantage of using form interfaces is that they are intuitive and easy to use. This paper presents research in progress on using natural language to fill out forms or any form-like input interface provided directly on the web or by the various available web wrapper technologies. Since web users of all levels are familiar and comfortable with forms, we will target our system to such interfaces. The form interface was chosen because the queries, though less powerful, are easier to formulate (just fill out the form) when compared to more traditional data retrieval languages such as SQL.

We feel there are several applications on the web that can benefit from a natural language interface. First, when the form must be large, typically it is broken up into a hierarchy of several forms. A natural language interface is not limited spatially, and thus can accommodate more information in one place. Second, people like to use natural language. This is proven by the popularity of the AskJeeves search engine, which accepts natural language questions. Third, a natural language interface for the web can support voice input more easily. When people speak, they want to use natural language. With a natural language back-end to a commercial dictation voice recognizer, people can gain better access to the web using speech.

We process natural language for filling out forms by spotting *field values* and using statistical approaches for disambiguating field value meanings. More specifically, a field value is a word or phrase that can be used as a value of a field in a form. For the simple form in Figure 1, the field values for the field "departure city" may be *Los Angeles*, *New York*, or *San Francisco*. Given a natural language user input, if each field value's corresponding field can be determined, then the form can be filled out. Consider the user input, "find me a flight from *Los Angeles* to *New York*." The field values, shown in italics, are *Los Angeles* and *New York* for the form of figure 1. If the system knows which field value corresponds to which field in the form, the form can be properly filled out. In this case, *Los Angeles* belongs to the "departure city" field and *New York* to the "arrival city" field. The field to which a field value belongs to is the field value's *meaning*. Because a given field value may have multiple meanings within a given domain, the field value must be disambiguated based on the lexical context in which it is used. Each field value will be associated with a list of possible meanings and statistical methods will be used to rank this list of possible meanings.

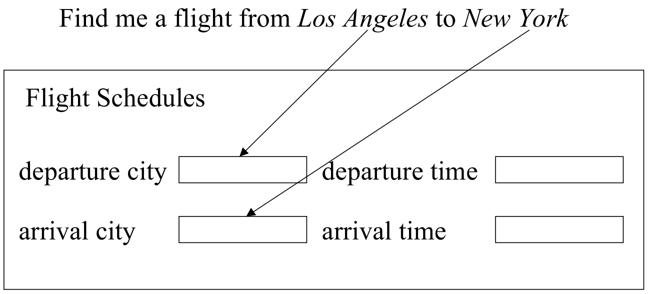

Figure 1. Simple Natural Language Interface Form

2. RELATED WORK

Natural language interfaces have been studied for the past several decades with many systems being developed enjoying varying levels of success. A good survey on the subject is found in Androutsopoulos, Ritchie and Thanisch 1995). Some foundational systems are described in Hendrix et al. (1978), Waltz (1978), and Woods (1978),and other later commercial systems are described in Androutsopoulos, Ritchie and Thanisch. Our work differs from most of these systems in that there is no need to parse the natural language input, which allows the system to be more portable and flexible.

Statistical language processing techniques typically use a corpus of natural language to compute statistics and then processes language based on these statistics. An overview of the field is found in Charniak (1993). Typical applications are information extraction (Costantino et al. 1997; Riloff and Lehnert 1994) and word sense disambiguation (Resnik and Yarowsky 1997; Wilks and Stevenson 1997; Yarowsky 1992). Of particular interest are the word sense disambiguation techniques, which determine the correct meaning of a polysemous word using statistics on context words or other sources. We found that these techniques, though very good at what they are designed to do, cannot be readily applied to disambiguating our field values. N-grams have been used in language models for predicting the nature of a word (e.g., its part-of-speech) using the last *n* words in the sentence (Charniak 1993). Letter-based n-grams have also been used in information retrieval (Mayfield and McNamee 1997).

There is much work on web wrappers that facilitate extracting data from web pages and some examples are found in Atzeni, Mecca and Merialdo (1997), Gruser et al. (1998), and Muslea, Minton and Knoblock (1999). Our system leverages on wrapper technology by providing a high-level interface to the data that they provide. Since we do not want to limit the scope of our system, it is not tied to any specific wrapper implementation.

3. METHODOLOGY

3.1 Choosing a Domain and Extracting Field Values

In this paper, a domain is defined as a set of related forms. For instance, a travel domain may be composed of forms for retrieving information on flight schedules from various airlines, hotel reservations, or rental cars. Once the domain has been established, the set of field values must be extracted. Forms for data retrieval have a relatively simple structure, consisting of an identifier, a set of fields and a set of values for each field. To take HTML as an example for expressing a form, the `<form>` element is used to represent the form. Fields are indicated by elements such as `<input>` or `<select>`, which represent text fields, select lists, etc. Any standard method can be used to parse HTML code to semiautomatically extract field values from an HTML form. Field values for select lists or choice menus can be directly extracted from the form. Values for fields such as text areas cannot be directly determined from the HTML code. A user of the form can determine what values are valid for the field because if they can use the form, they presumably understand the semantics of the form. Field values automatically extracted from a form may not be directly usable by the system because they may be internal, non-descriptive words. Again, we feel a user of the form can provide a mapping from these internal terms to their natural language equivalents.

3.2 Field Value Meanings and Statistical Disambiguation Techniques

One major problem when dealing with natural language is that words and phrases often have multiple meanings. Ambiguity arises in the proposed system when field values can refer to more than one field. For instance, in the travel domain, the field value *Los Angeles* may refer both to the "departure city" field or the "arrival city" field of a airline schedule form. A *meaning* of a field value is just the field it corresponds to. Field names will use the database-like convention of *form_name.field_name*. A field value will have associated with it a list of meanings, called its *meaning set*. Taking the same example, the field value *Los Angeles* may have the meaning set:

meaning_set(*Los Angeles*) = {airline.departure_city, airline.arrival_city}

Given a user input, a simple field value spotting algorithm can be used to determine the occurrence of any field values, and for each field value, its meaning set is derived. N-grams will be used to determine the best meaning for a field value from the possible meanings in its meaning set. N-grams are *n* contiguous words within a natural language text. They are easily generated by sliding a window of *n* words wide across the text, where each position of the window represents one n-gram. The digrams ($n = 2$) for the sentence "the black cat slept" are (the black), (black cat), and (cat slept). To determine the best meaning for a field value, the context in which the field value is used must be considered. To capture this context, a set of n-grams generated from the sentence containing the field value is used. A vector model of contexts will be used where each context will be represented by a vector of n-grams called the *context vector*. To determine the best meaning of the field value *Los Angeles* in the user input, "find me a flight from *Los Angeles* to *New York,*, the vector of n-grams generated from the user input will be the context vector for *Los Angeles*.

Each meaning (form field) in the domain will have a corresponding n-gram vector $\vec{V}_m$, defined as its *meaning context vector*, which represents the context in which the meaning is used. The meaning context vector of meaning *m* is an aggregate of all the contexts in which *m* was used as the meaning for a field value. To precompute a set of meaning context vectors to be used in a domain, a set of questions typical to the domain can be used. Each question is scanned for field values and when a field value is encountered in a question, its meaning is determined, and the context vector for the question is added to the meaning context vector of the meaning using vector addition. If meaning context vectors have been computed for each meaning in the domain, then the goodness measure *G(m)* of meaning *m* for a field value in the context of user input *I* is shown in equation 1, where $\vec{V}_I$ is the context vector for *I* and $\vec{V}_m$, is the meaning context vector for *m*.

$$G(M) + \cos(\vec{V}_I, \vec{V}_m)$$

Equation 1

Vector comparisons are done using the cosine of the angle between the two vectors, a standard measure used in information retrieval (Salton 1989). The goodness measure of a meaning is assigned to the cosine value and the meaning that has the highest goodness measure is deemed to be the meaning of the field value.

3.3. N-gram Experiments

We ran several experiments to verify the usefulness of n-grams in disambiguating field value meanings. We assumed that the domains were limited and that each field value had three or more possible meanings within the domain. Due to space limitations, we will only present the results of experiments on financial articles. In the financial articles, each numeric value was considered a field value with several possible meanings. The possible meanings for a numeric value are: the Dow Jones Industrial average points total, points change, and percentage change; the Nasdaq points total, points change, and percentage change; the NYSE total volumes, number of decliners, and number of advancers; and the S&P points total, points change, and percentage change. Six runs were made, using unigrams, digrams, trigrams, digrams with diminishing weights, trigrams with diminishing weights, and a Word Sense Disambiguation (WSD) technique. "Diminishing weights" assigns a weight to each n-gram based on its lexical distance from the field value. It was found that diminishing weights helped because the n-grams closest to the field value usually matter the most. The system was trained incrementally over five steps to show if the accuracy improves as more statistics are used. The WSD technique used for comparison is the one described in Yarowsky (1992), which determines a word's meaning based on its surrounding context words. A weight is assigned to each context word based on the probability it co-occurs with the word and how often it occurs in the corpus.

Figure 2 shows a graph of the six runs for the financial articles, where accuracy is shown versus the number of articles used to train the system. Each article contained roughly 10 field values. Digrams with diminishing weights performed the best, reaching almost 85% accuracy with only 50 articles of training. For most of the n-grams, the system also showed that more training led to better accuracy. These results show that n-gram statistics are effective for disambiguating field value meanings within a limited domain. The WSD method was not as effective because the context in which a field value is used for a certain meaning does not differ much from the context in which the same word is used with a different meaning. Because the WSD method relies on differing contexts to differentiate between several meanings of a field value, it does not have much to go by in a limited domain. Since n-grams take into account groups of words, it is able to capture the shallow structure of natural language, which has been noted to be useful in processing language (Soderland and Lehnert 1994). N-grams also take seemingly insignificant words (e.g., prepositions) into consideration, which have been shown to make a difference when processing language (Riloff 1995).

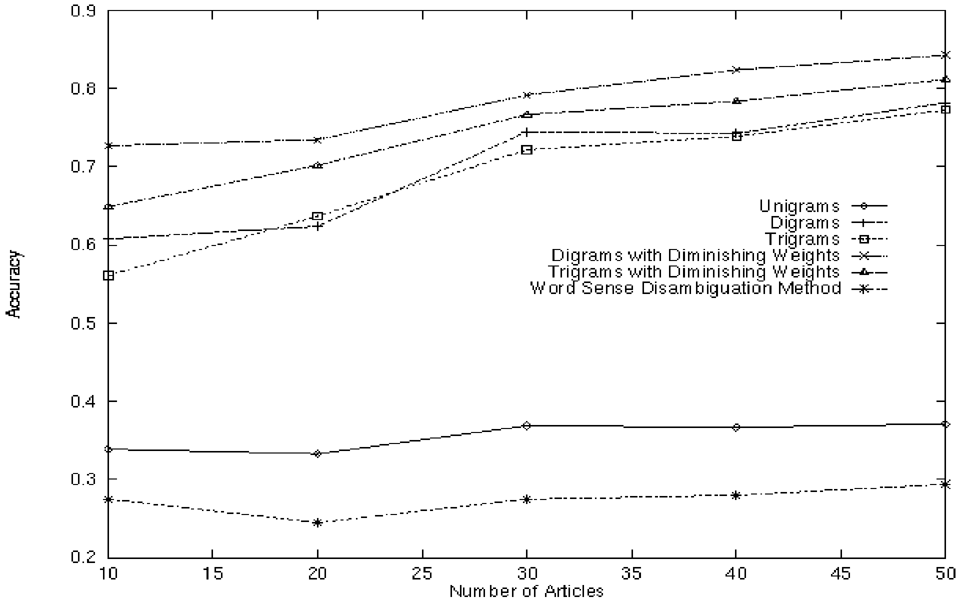

Figure 2. Six Runs of the Financial Articles

4. DISCUSSION

The main concern for scaling the system to large domains is with field value meaning disambiguation, which may degrade in performance as the domain increases in size. In actuality, it is not the general size of the domain that concerns the system, because a domain can contain millions of forms, and yet if each field value is unambiguous, the system will have no problem filling out forms. The performance of the system in a domain is directly related to the average number of possible meanings per field value in the domain. If each field value has many possible meanings, the system is more likely to make errors. We envision that many interfaces will be used in many small domains, as opposed to having one interface that handles the entire web.

Porting the system to a new domain involves determining what forms will participate in the domain, extracting the field values from each form, and building up the statistics needed for disambiguation. We have proposed a back engineering technique where existent forms are used to extract the necessary knowledge needed by the system. An alternative would be to provide web authoring tools that allow the form designer to "natural language-enable" his or her form. We feel web authoring tools in this case restrict the web page designer and forces providing additional information that will not be used. We also feel that back engineering a form does not incur much overhead in terms of effort because of the fact that forms are designed to be easy to use. Any proper user of a form will possess enough knowledge to be able to natural language-enable that form because he or she would know what the form represents, what each field represents, and would have a general idea of the valid values for each field.

The usability of the interface is dependent on the accuracy of the system and how much more work the user needs to do when system errors occur. A good measure of usability is the total time spent by the user to obtain the desired information. We plan to perform usability experiments with actual users over the web and compare the time per query for our interface with the original form interface.

5. CONCLUSION

This paper discussed some preliminary work done on a natural language interface to information access through forms on the world wide web. Form-based information access is a prevalent method for servers to make their data available to the general public. Forms are easy to use, clearly show what can be retrieved, and limit what can be retrieved. The system presented in this paper leverages on the prevalence of forms on the world wide web and their structured approach to information retrieval. Form queries are simple to generate and only require the system to determine what form to use and what values to fill into what fields of the form.

An approach based on field values and statistical field value meaning disambiguation techniques was described for processing a natural language user input and filling out the proper form. Field values can be semiautomatically extracted from a domain of forms and each field value has associated with it a list of possible form fields to which it can correspond (meanings). When a user input is accepted by the system, the field values are spotted and statistical disambiguation processing based on n-grams is used to determine the best meaning of each field value.

6. REFERENCES

Androutsopoulos, G.; Ritchie, D.; and Thanisch, P. "Natural Language Interfaces to Database Systems – An Introduction," *Journal of Natural Language Engineering* (1), 1995, pp. 29-81.

Atzeni, P.; Mecca, G.; and Merialdo, P. "To Weave the Web," in *Proceedings of the Twenty-third International Conference on Very Large Databases* (VLDB'97), 1997.

Charniak, E. *Statistical Language Learning*, Cambridge, MA: The MIT Press, 1993.

Costantino, M.; Morgan, R. G.; Collingham, R. J.; and Garigliano, R. "Natural Language Processing and Information Extraction: Qualitative Analysis of Financial News Articles," in *Proceedings of the Conference on Computational Intelligence for Financial Engineering (CIFEr'97)*, New York, March, 1997.

Gruser, J-R.; Raschid, L.; Vidal, M. E.; and Bright, L. "Wrapper Generation for Web Accessible Data Sources," in *Proceedings of the Third IFCIS International Conference on Cooperative Information Systems*, 1998.

Hendrix, G. G.; Sacerdoti, E. D.; Sagalowicz, D.; and Slocum, J. "Developing a Natural Language Interface to Complex Data," *ACM Transactions on Database Systems* (3:2), 1978, pp. 105-147.

Mayfield, J., and McNamee, P. "N-Grams vs. Words as Indexing Terms," in *TREC-6 Conference Notebook Papers*, November 1997.

Muslea, I.; Minton, S.; and Knoblock, C. "A Hierarchical Approach to Wrapper Induction," in *Proceedings of the Third Conference on Autonomous Agents*, 1999.

Resnik, P., and Yarowsky, D. "A Perspective on Word Sense Disambiguation Methods and Their Evaluation," in *Proceedings of SIGLEX'97*, Washington, DC, 1997, pp. 79-86.

Riloff, E. "Little Words Can Make a Big Difference for Text Classification," in *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995, pp. 130-139.

Riloff, E., and Lehnert, W. "Information Extraction as a Basis for High-Precision Text Classification," *ACM Transactions on Information Systems* (12:3), July 1994, pp. 296-333.

Salton, *G. Automatic Text Processing*, Reading, MA: Addison-Wesley Publishing Company, 1989.

Soderland, S., and Lehnert, W. "Wrap-Up: a Trainable Discourse Module for Information Extraction," *Journal of Artificial Intelligence Research* (2), 1994, pp. 131-158.

Waltz, D. L. "An English Language Question Answering System for a Large Relation Database," *Communications of the ACM* (21:7), 1978, pp. 526-539.

Wilks, Y., and Stevenson, M. "Combining Independent Knowledge Sources for Word Sense Disambiguation," in *Proceedings of the Third Conference on Recent Advances in Natural Language Processing Conference (RANLP-97)*, 1997, pp. 1-7.

Woods, W. A. "Semantics and Quantification in Natural Language Question Answering," in *Advances in Computers*, Volume 17, M. Yorvitz (ed.), New York: Academic Press, 1978.

Yarowsky, D. "Word-sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora," in *Proceedings of COLING-92*, 1992, pp. 454-460.