Page Ranking Refinement Using Fuzzy Sets and Logic

Andrew Laughlin **Joshua Olson** **Donny Simpson** **Atsushi Inoue** *
Eastern Washington University
Cheney, WA 99004 USA

Abstract

This paper presents a study on personalized add-on filters applied to web search results in order to make those results more intuitive to users. Fuzzy Sets and Logic are used in order to construct such filters. Linguistic features are extracted as their universe of discourse. Three experimental filters are presented in the following specific contexts: (1) narrowing down results, (2) product specification and (3) tutorial level classification. Their performance is briefly studied mostly in qualitative manners.

Keywords: Fuzzy Sets, Fuzzy Logic, Page Ranking, Linguistic.

Introduction

Users on the Internet very likely use search engines such as *Google* and *Bing* in order to search and access to information of interest. As they enter some key words, the search engines instantaneously respond lists of web pages that are *relevant* to those key words in the order of *significance*. While the quality of search results is generally satisfactory, the users often demand finer tunings, e.g. in terms of contexts, descriptors and dependencies among key words in phrases. The following lists a few examples of such discrepancies:

Contexts. Totally different contexts intended by users, e.g. a type of coffee beans vs. software components for the key word 'java beans'.

Descriptors. Descriptors that do not symbolically match with words in target web pages, e.g. some quantifiers such as 'most' vs. actual quantities such as '99%' and qualifiers such as 'good' vs. similar ones such as 'high quality' and 'well written'.

Dependencies among words. Symbolic keyword matching in search engines is most likely performed based on regular grammars (to handle word conjugations) and often yields weak or no relevancy, e.g. 'fuzzy parser', when interpreted as a list of two key words 'fuzzy' and 'parser', vs. texts in target web pages such as 'The parser avoids fuzzy words...', 'Parsing a query e.g. founder of fuzzy logic' and '... fuzzy sets. ... C parser to compile ...'.

In theoretical aspects, web page ranking predominantly follows a fundamental ingredient in the development and success of the *Google* search engine, and the significance is determined based on references and citations (i.e. links) made to that web page (a comprehensive survey is made in reference (Franceschet 2010)). The relevancy of a search result for given keywords is determined by this method applying to web pages containing those key words (with variations based on their conjugations)[1]. Such a ranking method is effective and efficient regardless of structures and contexts of target web pages and is indeed satisfied by many users despite the above mentioned discrepancies.

On the other hand, many others hope for additional fine tunings on the search results in order to overcome those discrepancies. As having been already noticed, all of those discrepancies are caused generally by lack of various linguistic processing on target web pages–e.g. lexical and semantic processing for the matters of contexts and descriptors, and syntactic and morphological processing for those of the dependencies among words. Knowing this difference on the basis of determining significance, i.e. structural (links) versus linguistic (texts), we anticipate an effectiveness of some linguistic functionalities that compensates the structural page ranking methods.

In this paper, we consider a study on such linguistic functionalities as *personalized add-on filters* that alter web page rankings generated by conventional web engines, e.g. *Google* and *Bing*, in specific, personalized contexts. The input of such a filter is a list of (links to) web pages in an order of significance based on their structures. In text processing aspects, this is considered as stream processing of texts with a demand of real-time response (e.g. just as Google responds to a search query). We deploy Fuzzy Sets and Logic as the base method given its proven efficiency on stream processing (e.g. Fuzzy Controls (Mamdani and Assilian 1975; Takagi and Sugeno 1985)) and effectiveness on uncertainty management intrinsic to linguistic processing (Zadeh 1965; 1973).

Related concepts such as page ranking and fuzzy sets and logic are briefly introduced in the next section. Then three experimental filters are presented along with their qualitative studies on performance for the following specific contexts:

*Correspond via E-mail to `inoueatsushij@gmail.com`

[1] In practice, massive web pages and their ranks are precompiled and key words are indexed by web crawlers, autonomous processes that explore links and URLs.

narrowing down results, product specification and tutorial level classification. Some clear distinction from other similar works is made within the experimental setting.

Related Concepts

The following related concepts are briefly introduced: Page Ranking, Regular Grammar, Fuzzy Sets and Fuzzy Logic.

Page Ranking

Let $I(P)$ the significance of page P and B be the set of pages that refer to (i.e. have a link to) page P (Austin 2011). Suppose that page P_i has a link to page P and that P_i has l_i links. Then $I(P)$ is determined as follows:

$$I(P) = \sum_{P_i \in B} \frac{I(P_i)}{l_i} \quad (1)$$

To determine $I(P)$, we need $I(P_i)\ \forall P_i \in B$ and so do we for each and every one of those pages. This certainly causes "chicken and egg" situation. To resolve this situation, we use the *power method*. Let $H = [H_{ij}]$ be a square matrix[2] representing references (links) among all web pages $P_1, \ldots, P_n$ such that

$$H_{ij} = \begin{cases} \frac{1}{l_j} & \text{if } P_j \in B_j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

and let $I = [I(P_i)]$ be a vector whose components are the significance of all the web pages. Then we observe $I = H \cdot I$. This means that I is an eigenvector (aka a stationary vector) with $\lambda = 1$. The *power method* iteratively computes

$$H \cdot I_i = v = \lambda_{i+1} \cdot I_{i+1} \quad (3)$$

until $I_i = I_{i+1}$ (i.e. practically $|I_i(P) - I_{i+1}(P)| < \epsilon$ for all components). The initial eigenvector usually has only one component whose value is 1, and the remaining components have 0s. The initial eigenvector is $\lambda_1 = 1$. The convergence of this iteration is determined whether $|\lambda_2| < 1$, and its speed (i.e. #iterations) is determined by the magnitude of $|\lambda_2|$ such that it gets slower as $|\lambda_2|$ is closer to 0.

Regular Grammar

Regular grammar can describe a set of strings by a collection of rules in the following patters: $A \rightarrow c$ and $A \rightarrow cB$ (Sipser 2005). Such rules yield only linear parsing trees. In practice such as scripting and programming, we use *regular expressions* that consists of the following:

|: Boolean "or."

$(\ldots)$: A regular expression within the parentheses.

$*$: Zero or more repetition of the preceding element.

$+$: One or more repetition of the preceding element.

?: Zero or one repetition of the preceding element.

For example, $ab?a$ yields aa and aba; $ab * a$ yields aa, aba and $abba$; $a(cb)+$ yields acb and $acbcb$; and $a(c|b)+$ yields ac, ab, acc, abb and abc.

[2] i-th row and j-th column.

Fuzzy Sets

In the most cases, fuzzy sets represent linguistic expressions that intrinsically contain fuzziness such as '*tall*' on height and '*low*' on leftover stipend (Zadeh 1965).

Definition. A *fuzzy subset* of a set U is defined by means of a membership function

$$\mu : U \rightarrow [0, 1] \quad (4)$$

The set U is so-called the *universal set*. In the above two examples, linguistic terms '*tall*' and '*low*' correspond to fuzzy sets defined over appropriate universal sets such as 'height' (i.e. a set (interval) of real numbers representing human height, e.g. $[100, 220]$ in centimeters) and 'leftover stipend' (e.g. $[0, 100]$ in USD). In case of a crisp set, the range of the membership function becomes $\{0, 1\}$ instead of $[0, 1]$.

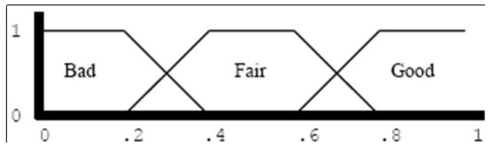

Figure 1: Fuzzy Sets (Fuzzy Partition)

Usually, fuzzy sets are defined in simple canonical shapes such as triangles and trapezoids (see fig. 1 as an example of three trapezoidal fuzzy sets). In this simplicity, you may easily see *the elasticity of fuzzy sets* such that every crisp (i.e. non-fuzzy) interval has only one fuzzy set i.e. $\mu(x \in U) > 0$ and $\mu(x) = 1$ (i.e. the complete membership), and every fuzzy interval has more than one fuzzy set i.e. $\mu(x) > 0$ and $\mu(x) < 1$ for all within that interval (i.e. the partial memberships). Further, a fuzzy partition is often considered for the sake of completeness in computational models.

Definition. A *fuzzy partition* of a set U is a set of normal (i.e. at least one element $x \in U$ s.t. $\mu(x) = 1$) fuzzy sets of U such that

$$\sum_i \mu_i(x) = 1\ \forall x \in U \quad (5)$$

Fuzzy sets may be defined subjectively, unlike probability distributions. Appropriate fuzzy partitions with simple canonical shapes as shown in fig. 1 are often used in many cases. They are also dynamically generated or refined by applying some machine learning methods. In such cases, simple and smooth shapes of fuzzy sets should be maintained due to their elasticity and approximation nature.

Finally, their *(standard) set operations* are defined as follows:

$$\begin{aligned} \textbf{Set intersection:} \quad & \mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] \\ \textbf{Set union:} \quad & \mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] \\ \textbf{Set complement:} \quad & \mu_{\overline{A}}(x) = 1 - \mu_A(x) \end{aligned} \quad (6)$$

Fuzzy Logic

Fuzzy logic is originally proposed by Zadeh as a qualitative, simplistic method for (especially complex) system analysis (Zadeh 1973). In this framework, Modus Ponens is generalized and formalized as a fuzzy relation (i.e. considered as a partial truth maintenance system). Formally the generalized Modus Ponens can be written as

$$a \to b \wedge a' \Rightarrow b' \tag{7}$$

where a, b, a' and b are fuzzy (sub)sets representing fuzzy statements, e.g. 'temperature is *high*'. We may now rewrite this in fuzzy set theoretic, i.e. fuzzy relational, aspects such that

$$\mu_{b'}(y \in V) = \bigvee_x [R_f(x, y) \wedge \mu_{a'}(x \in U)] \tag{8}$$

where all the membership functions μ represent those fuzzy statements (e.g. μ_{high} in the above example), U and V represent their universal sets (e.g. 'temperature' in the above example), and R_f is the fuzzy relation that represents the (fuzzy) implication $a \to b$.

That fuzzy relation is further specified as a result of projecting material implication (i.e. $a \to b = \neg a \vee b$) such that

$$R_f(x, y) = (a \times b) \cup (\overline{a} \times V) = (\mu_a(x) \wedge \mu_b(y)) \vee \mu_{\overline{a}}(x) \tag{9}$$

where $a \subseteq U$ and $b \subseteq V$.

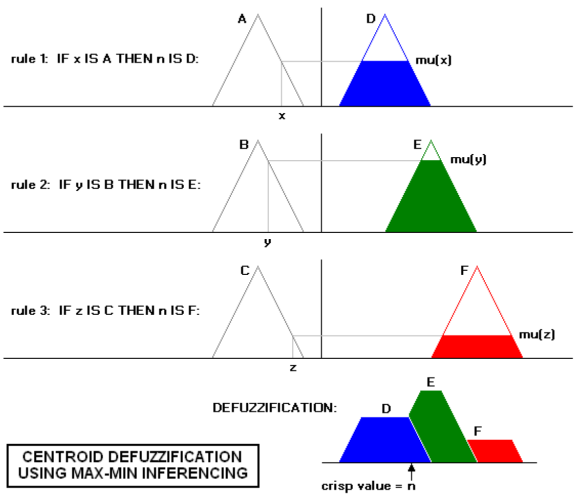

Figure 2: Fuzzy Control (Mamdani)

In fuzzy control, we only need to consider the special case[3] such that

$$R_f(x, y) = (a \times b) \cup (\overline{a} \times \emptyset) = (\mu_a(x) \wedge \mu_b(y)) \tag{10}$$

This is indeed Mamdani's fuzzy control model when selecting $\mu_a(x) \wedge \mu_b(y) = min[\mu_a(x), \mu_b(y)]$ (Mamdani and Assilian 1975) (see fig. 2).

When multiple fuzzy implications (aka fuzzy IF-THEN rules) exist in the system, we need to disjunctively combine all the results, i.e. partial truth such that $c = c_1 \vee \ldots \vee c_n$, where $c_{1 \le i \le n}$ is a fuzzy set representing the result for the i-th fuzzy implication and c is the one for all results combined. In case of fuzzy control, c represents all possible outputs with associated partial truth values. In order to determine a single output, we need to select one output such that $c(\subseteq V) \to y^*(\in V)$ (so-called *defuzzification*). Center-of-Gravity (CoG) method is proposed in Mamdani's model (see fig. 2).

$$y^* = \frac{\int \mu_c(y) \cdot y \; dy}{\int \mu_c(y) \; dy} \tag{11}$$

Takagi-Sugeno fuzzy control model better integrates fuzzy implications and defuzzification as a model-free regression (Takagi and Sugeno 1985). In this model, each fuzzy implication $R_f(x, y)$ is approximated as a function $y_i = f_i(x)$ for input x, and its output y_i is linearly combined based on the partial truth value of the hypothesis, i.e. $\mu_{a_i}(x)$, such that

$$y^* = \frac{\sum_i \mu_{a_i}(x) \cdot f_i(x)}{\sum_i \mu_{a_i}(x)} \tag{12}$$

In discrete problem domains such as classification, we need to identify a class as a result of disjunctively combining all those result fuzzy sets c_i. Since their membership functions are all constants (i.e. $\mu_{c_i}(y) = z_i \in [0, 1]$ s.t. $z_i = \mu_{a_i}(x)$ from fuzzy implication $a_i \to b_i$), the defuzzification is achieved simply as a result of the disjunctive combination with $max[\cdot]$ in order to select the class label. This is corresponding to the definition of fuzzy classifier such that

$$C = \text{ARGMAX}_i[\mu_{a_i}(x)] \tag{13}$$

where C is a class label associated with μ_{b_i} from fuzzy implication $a_i \to b_i$, as well as its result μ_{c_i}.

Experimental Filters

Three experimental filters are presented along with brief studies on their performance for the following specific contexts: narrowing down results, product specification and tutorial level classification. The experimental setting is presented first together with some clear distinction from other similar works.

Experimental Setting

In general, we consider *simplicity* as the core of development. In particular, the following setting is followed in the development of experimental filters.

Add-on Filters. Given all possible bias on intentions and interpretations, we focus on development of *personalized add-on filters* on web browsers. Such filters are very likely implemented as extensions and other forms of modules according to architectural specifications of web browsers as well as application programming interfaces (APIs). Fuzzy Sets and Logic are used as the technical framework of those filters and are easily implemented in any forms of development environment, application framework and programming language. The *input* of each filter is *a list of web pages*, most likely that of URLs, generated as a result of using a web search engine such as Google and Bing. The filter then accesses to texts from that list. The *output* is *a modified list*

[3] System control cannot specify outputs for complements of inputs, i.e. $\overline{a} \times \emptyset = \emptyset$.

of those web pages, e.g. altered orders, selective lists and grouped lists.

In doing so, users can easily switch back and forth between the ordinary and this filtered search results. In addition, the inputs, i.e. keywords, remain the same in both options. This is different from other works that utilize fuzzy sets and logic in a similar manner. For instance, Choi's work (Choi 2003) incorporates linguistic processing features (using fuzzy sets and logic) directly to a web search engine, thus demands a modification on the server as well as in inputs. This causes substantial overheads on the server including, but not necessarily limited to, configurations of various personalization and context dependencies. Such configurations may likely serve as very critical overheads when considering recent studies on bias and ideal usage of web search engines (Goldman 2008).

Recent search engines such as Google and Bing keep track of search results for various personalization and customization purpose. They are implemented as a part of server (i.e. search engine) functionalities. In contrast, ours are implemented as extensions of web browsers, thus are served as *additional personalization*.

Context Dependency. Each user has one's own intention and bias in many different situations and none are likely identical to the others. In other words, it is ideal (i.e. the simplest) to facilitate a collection of add-on filters that cater such different situations with various intentions and biases.

While many works in intelligent systems tend to handle context dependencies by adaptive capacity on the server, this causes substantial overheads. Anari et. al. approach to context dependencies by incorporating capacities of adaptive behaviors and generalization (i.e. capacity of fuzzy sets and logic) directly in the page ranking method (Anari, Meybodi, and Anari 2009). Such sophistication very likely causes the overhead, thus is not feasible for extensions of conventional services such as Google and Bing (unlike their own retrieval system).

Linguistic Keyword Processing. Fuzzy sets and logic are well incorporated with the standard *statistical natural language processing* such that simple features are extracted from texts in order to apply various methods of machine learning and reasoning. Such simple features may include *word frequencies* and *word appearances* as the core. We then consider other variations within those features such as

- those on words that are linguistically related, e.g. synonyms, antonyms, acronyms, etc.
- those on words with simply conjugation processing, more generally processing morphological structures.
- those on a sequence of n words, i.e. n-grams.

The most significant advantages of such simple features are with regard to text stream processing. Texts are parsed only once (aka one-pass) in order to yield real-time responses. Morphological structures are mostly handled by regular grammars (expressions), and those linguistically related features demand lexicons such as thesauri. Fortunately, those are feasible within the text stream processing framework. Conventional syntactic analysis is unlikely feasible within this framework; however, stochastic and probabilistic analysis on n-grams (e.g. Markov process) and simple parsing with regular grammar may often compensate for this shortcoming.

It is commonly known that the web search inputs, i.e. keywords, are short and simple–consisting only of a few keywords. As a consequence, a canonical structure such as a pair of an adjective and a noun can be expected. This very well fits within the frame work of fuzzy sets and logic such that the adjective (a word or a phrase) corresponds to a fuzzy subset on the universal set and the noun (a word or a phrase) corresponds to that universal set. The elements in that universal set are one of those simple linguistic features and the fuzzy set is generated accordingly on those.

Prototype. All the experimental filters presented in this paper are required to be rigorously prototyped. The web search results are indeed extracted from Google and Bing for specific queries, i.e. lists of keywords. The standard models of fuzzy sets and logic are deployed, e.g. the standard fuzzy set operations and Mamdani's fuzzy control model, for implementation advantages such as simplicity and available tools. As a trade-off, performance studies became very limited at this time–i.e. a continuing work.

Filter 1: Narrowing Down Results

This experimental filter considers narrowing down and reordering search results from a web search engine about favorite music. The intention is exploration of relevant information, thus is broad and general. More technical details are as follows:

Queries. For this experiment, we only consider "good reggae songs."

Features. The universal set is the normalized frequency $f = \frac{f(w \in t \wedge d)}{f(w \in t) - f(w \in t \wedge i)}$ of affirmative words in a text t, e.g. 'good', 'favorite', 'love', etc. Function $f(w)$ indicates the frequency of word w with a specified membership (e.g. t, $t \wedge d$ and $t \wedge i$). A thesaurus d is used in order to identify a set of words and their conjugations. We remove meaningless words such as prepositions and pronouns and they are maintained in the ignored word list i.

Fuzzy model. A fuzzy classifier consisting of three fuzzy sets in a fuzzy partition similar to those in fig. 1: $\mu_{\text{high}}(f(t))$, $\mu_{\text{medium}}(f(t))$ and $\mu_{\text{low}}(f(t))$.

Output. (1) a class of the word frequency as a fuzzy degree of significance, and (2) a defuzzification of those three membership functions as the degree in order to determine the rank.

Search engine. Bing.

A small and simple performance evaluation was conducted as follows:

Subjects to evaluation. Significance labels generated by the fuzzy classifier.

Examinee(s). One person who is familiar with reggae songs.

Table 1: Confusion Matrix

M \ H	High	Medium	Low
High	1	2	0
Medium	0	3	0
Low	1	0	2

Procedure. Several web pages (texts) that are randomly selected from the filtered results are presented to the examinee and ask the one to put one of those three labels ('low', 'medium' and 'high') in order to indicate significance.

Results. Compiled as a confusion matrix (see table 1) where H indicates the human examinee and M indicates this add-on filter.

Despite its very simple linguistic feature, i.e. the normalized word frequency of single word, we noticed some improvements. Two out of three misclassification cases are classified in adjacent classes: 'high' where it should be 'medium'. One case is completely off; however, this was the web page containing a song list and hardly contained affirmative words.

Table 2: Top 5 Results from Bing

Bing Ordering	Fuzzy Value	Human Value	Reason
1	Medium	Medium	Forum suggestion.
2	Low	High	Long list of songs. Few other terms.
3	Medium	Medium	Forum suggestion.
4	Medium	Low	Short list of links. Non-familiar artists.
5	Medium	Medium	Forum suggestion.

Table 2 indicates the results of the top 5 from Bing search. Two pages are off, but both are song lists and hardly contain affirmative words. Needless to say, more extensive studies are necessary. Nevertheless, this small performance study positively suggested further study.

Filter 2: Product Specification

This experimental filter anticipates to enhance product search results. The key idea is that a certain specification is accounted to significance of products appeared in the search results. Such a specification is automatically identified as a result of mapping a set of key words to some fuzzy model. The technical details of this filter follows:

Queries. For this experiment, we only consider "energy efficient light bulbs."

Features. The universal set is the efficiency $e(b) = \frac{l(b)}{w(b)} \in [5, 100]$, where $l(b)$ is the lumen and $w(b)$ is #watts of a light bulb b. Product specification such as $l(b)$ and $e(b)$ are obtained from Google's product search results by parsing XML attributes and keywords.

Fuzzy model. A single fuzzy set representing 'energy efficiency' such that

$$\mu(e(b) \in [5, 100]) = \begin{cases} 0 & \text{if } e(b) \in [5, 15] \\ \frac{e(b)-15}{35} & \text{if } e(b) \in (15, 50) \\ 1 & \text{if } e(b) \in [50, 100] \end{cases} \tag{14}$$

This is determined based on the official chart of Energy Federation Incorporated (EFI).

Output. Reordered list of products according to the membership degree of $\mu(e(b))$.

Search engine. Google Shopping (product search service).

A qualitative performance study on this filter is conducted as follows:

Subjects to evaluation. Quality of product search results.

Examinee(s). A few people who are in need of light bulbs.

Procedure. Obtain testimonials by presenting both product lists: filtered and not-filtered.

Results. A collection of testimonies are listed below. See fig. 3&4 to compare top 8 products (i.e. energy efficient light bulbs). Five bulbs in the filtered results also appear in the not-filtered one. Among those, only one (the top bulb) appear in the same rank. Three new bulbs appear in the filtered list.

Here is the sample testimonies:

1. "I think [the filtered result] is better it sticks more to sorting the wattage in order, making the *energy efficiency* stand out more if people are looking to cut costs by reducing wattage. This is not taking things like price or actual lumenage of the bulb into effect (if people are looking for a brighter, yet energy efficient bulb, which might actually be a higher wattage)."

2. "I'd have to go with [the filtered result] mainly because it ranked the Satco Halogen among the lowest. The Satco Halogen uses the most energy out of all the bulbs with the least gain of lumen. 57 watts for 1100 lumens vs 13 watts for 900 lumens."

This filter, despite its simplicity in uncertainty management, turned out to be better effective than we expected. The current trends on ecology stimulated the bulb market so that manufacturers release new types of bulbs and users (i.e. consumers) are not yet familiar with those new types. More comprehensive implementation of such a filter is likely significant from marketing aspects and is certainly feasible from technical aspects.

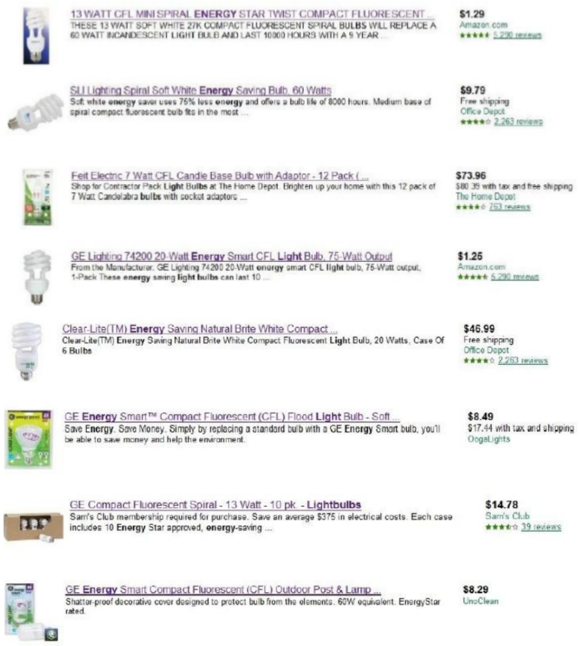

Figure 3: Product Search Results: not filtered

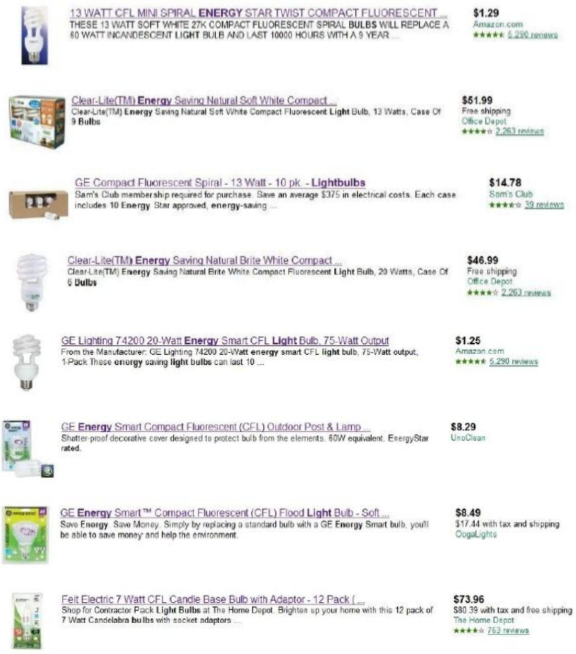

Figure 4: Product Search Results: filtered

Filter 3: Tutorial Level Classification

This experimental filter classifies tutorial sites into three classes: 'beginner', 'intermediate' and 'advanced'. Only a set of few general keywords are taken into account for this classification. In doing so, we maintain a high degree of generality (i.e. domain independence). Each of such keywords are associated with appropriate membership degrees per class. Mamdani's model is deployed for this classification as a result of defining singleton fuzzy sets over a shared universal set representing those classes.

Queries. For this experiment, we consider: $(c|ruby|python)(tutorials?|guides?)$

Features. We only use the following keywords and their conjugation (handled by regular expressions): $intro(a|\dots|z)*$, $novice$, $intermediate$, $advanc(a|\dots|z)*$, $experts?$. We also use their frequencies: $f_i(t)$ for $intro(a|\dots|z)*$, $f_n(t)$ for $novice$, $f_m(t)$ for $intermediate$, $f_a(t)$ for $advanc(a|\dots|z)*$ and $f_e(t)$ for $experts?$.

Fuzzy model. The following IF-THEN rules and fuzzy sets are defined:

1. $h(f_i(t)) \to c_i^h = \{1.0/0\}$
2. $h(f_n(t)) \to c_n^h = \{1.0/0.25\}$
3. $h(f_m(t)) \to c_m^h = \{1.0/0.5\}$
4. $h(f_a(t)) \to c_a^h = \{1.0/0.75\}$
5. $h(f_e(t)) \to c_e^h = \{1.0/1\}$
6. $m(f_i(t)) \to c_i^m = \{0.5/0\}$
7. $m(f_n(t)) \to c_n^m = \{0.5/0.25\}$
8. $m(f_m(t)) \to c_m^m = \{0.5/0.5\}$
9. $m(f_a(t)) \to c_a^m = \{0.5/0.75\}$
10. $m(f_e(t)) \to c_e^m = \{0.5/1\}$

where

$$h(f) = \begin{cases} 0 & \text{if } f \leq 5 \\ \frac{f-5}{5} & \text{if } 5 < f < 10 \\ 1 & \text{if } f \geq 10 \end{cases} \tag{15}$$

and

$$m(f) = \begin{cases} 0 & \text{if } f = 0 \vee f \geq 10 \\ \frac{f-5}{5} & \text{if } 0 < f < 5 \\ 1 & \text{if } f = 5 \\ \frac{10-f}{5} & \text{if } 5 < f < 10 \end{cases} \tag{16}$$

Output. The degree of expertise $e(t) \in [0, 1]$ as a result of defuzzification of this fuzzy model, where $e(t) = 1$ means 'expert' and $e(t) = 0$ means 'beginner'. The degree of each class is determined as follows:

Beginner: $b(t) = 1 - e(t)$

Intermediate: $i(t) = 1 - |e(t) - 0.5|$

Advanced: $e(t)$

Search engine. Bing and Google.

Fig. 5 and 6 show a search result and a classified result of Google with query "python tutorial." As you may notice, there are a few pages in unintentional context, i.e. Pokemon. This is a trade off with generality, that is caused due to a shallow linguistic analysis such as word frequencies of only a few keywords. Nevertheless, we received several positive feedbacks about this filter that mainly commends such a classification offers more efficient information browsing.

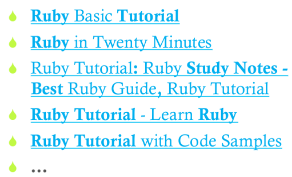

Figure 5: Search Results:Google

Figure 6: Classified Results:Google

Concluding Summary

A preliminary study on page ranking refinement using fuzzy sets and logic is presented. Three simply experimental filters are presented in order to demonstrate their effectiveness. Their performance studies, despite their rough and mostly qualitative contents, positively suggest continuing works. In experimental setting, clear difference from similar works is discussed.

Among many future works, we will first conduct more extensive performance studies and prepare some development frameworks for popular web browsers such as Fire Fox and Chrome.

Acknowledgement

This research was conducted as a small course project (so-called *Intelligent Search Project*) in CSCD498/598 Fuzzy Sets and Logic at EWU at the end of Fall 2010. The duration of this project was two weeks. The best three projects are presented as experimental filters in this paper and those students are recognized as its co-authors.

References

Anari, Z.; Meybodi, M. R.; and Anari, B. 2009. Web page ranking based on fuzzy and learning automata. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '09, 24:162–24:166. New York, NY, USA: ACM.

Austin, D. 2011. How Google Finds Your Needle in the Web's Haystack. *AMS Feature Column: Monthly Essays on Mathematical Topics*.

Choi, D.-Y. 2003. Enhancing the power of Web search engines by means of fuzzy query. *Decis. Support Syst.* 35:31–44.

Franceschet, M. 2010. PageRank: Standing on the shoulders of giants. *ACM Computing Research Repository (CoRR)* abs/1002.2858v3.

Goldman, E. 2008. Search Engine Bias and the Demise of Search Engine Utopianism. In Spink, A., and Zimmer, M., eds., *Web Search*, volume 14 of *Information Science and Knowledge Management*. Springer Berlin Heidelberg. 121–133.

Mamdani, E., and Assilian, S. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7(1):1 – 13.

Sipser, M. 2005. *Introduction to the Theory of Computation (2nd eds.)*. Course Technology (Thomson).

Takagi, T., and Sugeno, M. 1985. Fuzzy identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics*.

Zadeh, L. A. 1965. Fuzzy Sets. *Information and Control* 8:338–353.

Zadeh, L. A. 1973. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics* 3(1):28–44.