# Intelligent tagging of online texts using fuzzy logic

Robertas Damaševičius, Remigijus Valys
Department of Software Engineering
Kaunas University of Technology
Kaunas, Lithuania
robertas.damasevicius@ktu.lt

Marcin Woźniak
Institute of Mathematics, Faculty of Applied Mathematics
Silesian University of Technology
Gliwice, Poland
Marcin.Wozniak@polsl.pl

***Abstract*—We propose four fuzzy-logic based models of tag recommendation, which are based on the interpretation of word frequency as a fuzzy membership function, and provide experimental results of tag recommendation for a variety of text datasets using different fuzzy logic operators. The novelty of the proposed models is the use of fuzzy logic modeling concepts to define a set of tags, the use of an existing set of tags for the selection of tags to strengthen the selection of most relevant (i.e., commonly used) tags, and the possibility to use an ontology to select semantically generalized tags. A system developed using the proposed models is adaptive (adapts the recommended tags to the existing set of tags), has a feedback (after each tagging, the set of tags and the dictionary are updated), is personalized (each user develops its own set of tags), and is semantics-aware (uses an ontology to refine tags). The models are validated using five sets of texts with different topics (technology, cooking, carrier, scientific, nature) and different length.**

***Keywords—text tagging, fuzzy logic, recommender systems, semantic web, text mining.***

## I. INTRODUCTION

*Semantic Web* is a transformation of current web abundant with unstructured text or data into a web of data annotated with machine readable metadata (semantic description of data), which can be processed (directly or indirectly) by computers [1]. The aim of Semantic Web is to achieve intelligence, including stand-alone intelligence of web systems, which can reason and infer new knowledge based on the meaning of the data, and *collective intelligence* [2] that emerges from collaboration and competition of many individuals and leads to computer-based decision making. The examples include *crowdsourcing* based on the idea that the crowd (a loosely connected group of people) on average is more intelligent than an individual and can collect and aggregate large amounts of information to gain an accurate understanding of a topic [3]. Another form of collective intelligence is *social tagging* [4], when a community of users collaboratively creates and manages tags to annotate and categorize web content. Though usually users are free to choose any tag to annotate the content, the process creates a *folksonomy* [5], i.e., a common vocabulary shared by the community, which reflects similar semantic meanings of words and concepts, even when the tags vary across individuals. Such folksonomy can be further used to search against tagged resources and infer other knowledge on the annotated data and on the users themselves. Collective creation of folksonomies can provide a number of advantages such as increased personalization of human-computer interaction [6], improved communication in software development [7], cultural and social adaptivity of web content [8], increased intelligence of applications [9], and increased user satisfaction through engagement and gamification [10]. Already, social tagging has been proven to increase web search precision [11]. However, it is difficult to persuade users other than in niche communities such as music or cinema fans, to tag web content consistently (efforts include the use of gamification techniques [12], etc.).

Social tagging has emerged as one of the best ways of associating metadata with web content. Tags can be key words or key phrases attached to documents or web objects (blog entries, photos, music, or videos) to describe the meaning (or semantic properties) of these objects [13]. Different kinds of tags may be used such as content-based, context-based, attribute, ownership, subjective (affective), organizational, purpose-based, factual, personal, self-referential, geographical, etc. [4]. Usually, humans add tags to the information they consider relevant and that information becomes semantic web content. This can be done manually or software can help by recommending most popular or semantically relevant tags. However, the manual approach can fail to provide a consistent tagging especially when the size of the tag vocabulary increases and the semantic knowledge of users diverge.

Golder *et al.* [14] identified major problems with current tagging systems: *polysemy* (a single tag can have multiple meanings), *synonymy* (multiple tags can have the same meaning), and *level variation* (users tag content at differing levels of abstraction). Synonymy is especially challenging as posts on the same topic can be tagged by different tagsets yet they have the same meaning. Another factor that complicates tagging is the lack of pressure to make tagging consistent, and complete for use in IT applications dealing with collaboration, clustering, and search. Inconsistent tagging using tags unrelated to content may be used to make posts visible to a larger community or just for organizing posts for own consumption. Automated tagging systems could be useful by suggesting to the users the tags, which are more consistent, descriptive and meaningful. Also automatic tagging can provide an effective way to complete manually added tags, especially for dynamic or very large collections of documents [15].

The aim of this paper is to assign tags to texts automatically using the proposed fuzzy-logic based models. The novelty of the proposed models is 1) the use of fuzzy logic modeling concepts to define a set of tags, and 2) the use of an existing set of tags for the selection of tags to strengthen the selection of most relevant (i.e., commonly used) tags.

## II. RELATED METHODS AND TAGGING SYSTEMS

Automatic social tagging domain spans several areas of research including text pre-processing, lexical analysis, removal of stop words, stemming, term ranking, topic (or keyword) extraction, tag mining and tag recommendation. For stemming, Porter algorithm [16] is often used to extract word stem in English language. Perhaps the best known method for term ranking is TF–IDF, the product of two term frequency (TF) and inverse document frequency (IDF) [17]:

$$tfidf(w,d,D) = tf(w,d) \cdot idf(w,D) \quad (1)$$

here $tf(w,d)$ is the frequency of word $w$ in document $d$; $idf(w,D)$ is the inverse document frequency $idf(w,D) = \log \frac{N}{1+|\{d \in D : w \in d\}|}$; $N$ is the number of documents; and $D$ is the set of documents.

Different improvements of the TF-IDF measure exist such as TF*PDF [18], MIDF [19], Okapi [20], and LTU [21], which use additional parameters for term weighting.

*Domain Consensus* simulates the consensus that a term must gain in a community before being considered as a relevant domain term [22]. Domain Consensus captures domain concepts that exhibit high frequencies within a small subset of the corpus (e.g., single document), but are completely absent in the remainder of the corpus. The consensus is high if a term has an even probability distribution across the documents chosen to represent the domain. *Lexical Cohesion* evaluates the degree of cohesion among the words that compose a terminological string [23]. The cohesion is high if the words composing the term are more frequently found within the term than alone in texts. *Domain Pertinence* of domain corpus is a filter that checks whether a term is relevant for the target domain [23]. Domain Pertinence is high if a term is frequent in the domain of interest and much less frequent in other domains used for contrast. *Glossex* [24] measures 'termhood' by comparing term frequency in the target corpus with its frequency in a reference corpus; then measures 'unithood' based on the overall term frequency normalized with respect to the frequencies of the component words. *Structural Relevance* increases the weights of a term if it appears in the title or paragraph title, or if it is in bold or underlined, etc. [25].

A different term ranking method extracts terms based on their frequency of co-occurrence [26]. Frequent terms are extracted first, and then a set of co-occurrences between each term and the frequent terms, i.e., occurrences in the same sentences, is generated. Co-occurrence distribution shows the importance of a term in the document as follows. If probability distribution of co-occurrence between a term and the frequent terms is biased to a particular subset of frequent terms, then the term is likely to be a keyword. The degree of biases of distribution is measured by the $\chi^2$-measure as follows:

$$\chi(w)^2 = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \quad (2)$$

here $freq(w,g)$ is the co-occurrence frequency, $n_w$ is how often the word $w$ was repeated with the best term $g$; $p_g$ is how often the best term $g$ was repeated in the text.

Common computational and artificial intelligence methods such as Artificial Neural Networks (ANN), Latent Semantic Analysis (LSA) and Bayesian models can be used for tag selection. For example, Latent Dirichlet Allocation (LDA) is a generative three-level hierarchical Bayesian probabilistic model for collections of discrete data such as text documents [27]. The documents are modeled as a finite mixture over an underlying set of topics which, in turn, are modeled as an infinite mixture over an underlying set of topic probabilities. The topic probabilities provide an explicit representation of the documents. Song *et al.* [28] use spectral recursive embedding clustering and a two-way Poisson mixture model for real-time tagging of Web documents. A new document is classified by the mixture model based on its posterior probabilities so that tags are recommended according to their ranks. Lee and Chun [13] propose the aAT:tag algorithm for automatic tag recommendation for blogs that uses collective intelligence extracted from Web 2.0 collaborative tagging as well as word semantics to learn how to predict the best set of tags using a hybrid ANN. The method uses both statistical methods (TF-IDF, word co-occurrence frequency) and lexical resources (WordNet) for keyword extraction. Yang and Lee [29] convert text into a vector of terms and then use self-organizing maps (SOM) to cluster vectors. The method aims to measure the relatedness between a Web page and a tag and claims that that the intrinsic relatedness may be revealed by training a large amount of samples. Tsai [30] presents a tag-topic model for blog mining, which is based on the Author-Topic model and LDA. The tag-topic model determines the most likely tags and words for a given topic in a collection of blog posts. Krestel *et al.* [31] apply LDA to the dense core of a folksonomy to extract topics which are later used to recommend additional tags for infrequently tagged resources. Montanes *et al.* [32] propose an approach to collaborative tag recommendation based on a machine learning system for probabilistic regression. Platt and Platt [33] use Support Vector Machine (SVM) to obtain a probabilistic output for a binary classification, performing a separate classification for each tag (category) in the training set. But this becomes unfeasible in case of large collections with hundreds of thousands of tags. Elisseeff and Weston [34] propose a multi-label system based on SVM, which generates a ranking of categories. The drawback is that the complexity, which is too high to be applied to real data sets. Symeonidis *et al.* [35] used a generalization of Singular Value Decomposition (SVD) to model the relations between users, resources and tags. Each of such triplets is assigned a probability value. Given a user and resource, the system returns the most probable tags related to them, hence the recommendation process is very efficient. Fujimura *et al.* [36] propose a method of multi-autotagging, based on the k-NN classification method and propose the residual document frequency metric to score similarity between tags to perform the term weighting.

Different heuristic methods are also used to filter out generic modifiers, distinguish terminology from proper nouns,

detect misspellings (e.g., using WordNet), extract single-word terminology, detect acronyms, evaluate term relevancy, etc. The known commercial services for term extraction and tagging include TermExtraction (by Yahoo), TagCrowd, VisualText, Topicalizer, and OpenCalais.

Fuzzy logic modeling is, however, rarely used in the area of tag recommendation. We have managed to find only one publication: Al-Kofahi *et al.* [37] propose TagRec, an automatic tagging algorithm for textual artifacts that is based on the fuzzy set theory. For each word, TagRec defines a fuzzy set. Each item has a membership value in this set, with 0 corresponding to no membership in the set defined by the term, and 1 corresponding to full membership. To compute the membership values for all items with respect to all words in the corpus, TagRec first builds a correlation matrix for all meaningful words based on their co-occurrence. Then, the membership values are computed based on the principle that an item belongs to the fuzzy set associated to the word, if many of its own words are strongly related to the word. Finally, the words corresponding to the membership values that exceed a threshold are reported as the tags.

The proposed method is different from the Al-Kofahi *et al.* [37] model, first, by providing a different interpretation of the fuzzy membership that is frequency-based rather than correlation-based, and by incorporating different interpretations of fuzzy operators rather than a single one (Al-Kofahi *et al.* use only the algebraic interpretation).

## III. FUZZY LOGIC BASED MODELS OF TEXT TAGGING

### A. Basic concepts of fuzzy logic modelling

A fuzzy set A in U, the universe of discourse under discussion is identified by a membership function $\mu_A: U \rightarrow [0,1]$ defined such that for any element $x$ in U, $\mu_A(x)$ is a real number in the closed interval $[0,1]$ indicating the degree of membership of $x$ in A [38]. The fuzzy set theory defines fuzzy operators on fuzzy sets, which are an extension of Boolean logic operators for fuzzy set variables. Mathematically, such operators can be interpreted differently using, e.g., algebraic, Zadeh, Einstein, Hamacher, Dubois, Yager, Frank and Schweizer operators [39].

Using fuzzy variables, fuzzy logic operators and fuzzy inference, fuzzy logic rules can be defined. Fuzzy rules play a key role in representing expert control/modeling knowledge and experience in linking the input variables of fuzzy controllers/models to output variable (or variables).

Two major types of fuzzy rules exist: Mamdani (Type-2) fuzzy rules and Takagi-Sugeno (Type-3) fuzzy rules. The Mamdani fuzzy rules [40] can be defined using the Backus-Naur Form (BNF) notation as follows:

```
IF <fuzzy_condition> THEN <action>
<fuzzy_condition> ::=
<fuzzy_variable> [<fuzzy_oper><fuzzy_variable>]
|<fuzzy_value>
```

In the Takagi-Sugeno (TS) fuzzy rules [41], the output of the fuzzy rule is defined as a function rather than a linguistic term as follows (see the description in BNF):

```
IF <fuzzy_condition>
THEN <output> := <fuzzy_function>
```

Here we combine the ideas of word frequency based tagging with fuzzy logic models by interpreting word frequency as the degree of membership of a word in a tagset. We describe our proposed fuzzy logic based models for tag recommendation below.

### B. Model-F0: Text

Let fuzzy-logic tagging model M be a tuple $M = (L, W, \hat{T}, \lambda, \varrho)$, where L is a language over alphabet $\Sigma$, W is a finite set of words from language L, $W \subset \Sigma^*$, $\hat{T}$ is a finite set of tags extracted from W such that $\hat{T} \subset W$, $|\hat{T}| \ll |W|$, $\lambda: W \rightarrow [0,1]$ is a fuzzy membership function of a word which represents its frequency in a text W, and $\varrho: W \rightarrow [0,1]$ is a fuzzy membership function of a word-tag in a set of tags $\hat{T}$. The input of the model is the analyzed text, and the output is a set of tags $\hat{T}$, which is defined using the TS fuzzy rule:

IF $\lambda(w)$ IS "high" THEN

$\hat{T} = \sigma(W), \forall w_i \in \hat{T} \subseteq W, \nexists w_j \in W, \varrho(w_i) < \varrho(w_j), w_i \neq w_j$ and $|\widehat{T}| = 1 + \log_{10}|W|$,

here $\sigma$ is the relational algebra selection operator, and $\varrho(w) = \lambda(w)$.

The Model-F0 fuzzy tagging algorithm is as follows:

1. Parse text into words.
2. Calculate fuzzy word membership in the text.
3. Select top word-tags.

### C. Model-F1: Text and Dictionary

Model-F1 extends model-F0 with dictionary D, which contains information about a frequency of each word in a language L or a text corpus.

Let the fuzzy-logic tagging model M be a tuple $M = (L, D, W, \hat{T}, \mu, \lambda, \varrho)$, where L is a language over alphabet $\Sigma$, D is a finite set of common words (dictionary) in a language L, $D \subset \Sigma^*$, W is a finite set of words in a text consisting of words from language L, $W \subset D$, $\hat{T}$ is a finite set of tags extracted from W such that $\hat{T} \subset W$, $|\hat{T}| \ll |W|$, $\mu: D \rightarrow [0,1]$ is a fuzzy membership function of a word in a dictionary D which represents the frequency of its occurrence in a text corpus of a language L, $\lambda: W \rightarrow [0,1]$ is a fuzzy membership function of a word which represents its frequency in a text W, and $\varrho: W \rightarrow [0,1]$ is a fuzzy membership function of a word-tag in a set of tags $\hat{T}$. The inputs of the model are a dictionary D and a set of words W in a tagged text. The output of the model is a set of tags $\hat{T}$, which is defined using the following TS fuzzy rule:

IF $\lambda(w)$ IS "high" AND $\mu(w)$ IS "low"

THEN $\hat{T} = \sigma(W), \forall w_i \in \hat{T} \subseteq W, \nexists w_j \in W, \varrho(w_i) < \varrho(w_j), w_i \neq w_j$ and $|\widehat{T}| = 1 + \log_{10}|W|$

We provide the following heuristic definition of the 2-relational operator $\varrho$ as:

$$\varrho(w) = \lambda(w) \cap \overline{\mu(w)}$$

here $\lambda(w) = f(w, W), w \in W$, is the function of the frequency f of word w in the analysed text; $\mu(w) = \begin{cases} f(w, D), w \in D \\ 0, w \notin D \end{cases}$, here the function $\mu(w)$ is equal to 1 if the word w is not found in the dictionary D either because it is a rare word, or a new word (e.g., abbreviation) introduced by the user himself; otherwise it is equal to the frequency f of a word w in a text corpus represented by the dictionary D; $\overline{(\cdot)}$ is fuzzy complement operator, and $\cap$ is fuzzy intersection operator.

We have implemented the following tag fuzzy membership function using the interpretations of fuzzy union and fuzzy complement operators of Zadeh, Einstein, Hamacher, Dubois, Yager,Frank, Schweizer, bounded and algebraic [39]:

Algebraic: $\varrho(w) = \lambda(w) \cdot (1 - \mu(w))$

Bounded: $\varrho(w) = \max[0, \lambda(w) - \mu(w)]$

Zadeh: $\varrho(w) = \min[\lambda(w), 1 - \mu(w)]$

Dubois: $\varrho(w) = \frac{\lambda(w) \cdot (1 - \mu(w))}{\max[\lambda(w), 1 - \mu(w)]}$

Yager: $\varrho(w) = 1 - \min\left[1, \sqrt{(1 - \lambda(w))^2 + \mu^2(w)}\right]$

Schweizer: $\varrho(w) = \frac{1}{\sqrt{\frac{1}{\lambda^2(w)} + \frac{1}{(1 - \mu(w))^2} - 1}}$

Einstein: $\varrho(w) = \frac{\lambda(w) \cdot (1 - \mu(w))}{1 + (1 - \lambda(w)) \cdot \mu(w)}$

Hamacher: $\varrho(w) = \frac{\lambda(w) \cdot (1 - \mu(w))}{\lambda(w) + (1 - \mu(w)) - \lambda(w) \cdot (1 - \mu(w))}$

Frank: $\varrho(w) = \log_2\left[1 + (2^{\lambda(w)} - 1)(2^{1 - \mu(w)} - 1)\right]$

The Model-F1 fuzzy tagging algorithm is as follows:

1. Parse text into words.
2. Calculate fuzzy word membership in the text.
3. Calculate fuzzy word-tag membership using word-dictionary.
4. Select top word-tags.

### *D. Model- F2: Text, Dictionary and Tagset*

Model-F2 extends model-F1 with user tagging history, also called personomy, and employ tags previously posted by the user for recommendation as tags for a current resource.

Let the fuzzy-logic based tagging model M be a tuple $M = (L, D, W, T, \hat{T}, \mu, \lambda, \nu, \varrho)$, where L is a language over alphabet $\Sigma$, D is a finite set of common words (dictionary) in a language $L$, $D \subset \Sigma^*$, W is a finite set of words in a text written in language L, $W \subset D$, T is a finite set of tags such that $T \subset D$, $|T| \ll |D|$, $\hat{T}$ is a finite set of tags extracted from W such that $\hat{T} \subset W$, $|\hat{T}| \ll |W|$, $\mu: D \to [0,1]$ is a fuzzy membership function of a word in a dictionary D which represents the frequency of its occurrence in a text corpus of a language L, $\lambda: W \to [0,1]$ is a fuzzy membership function of a word which represents its frequency in a text W, $\nu: T \to [0,1]$ is a fuzzy membership function of a tag which represents its frequency in a set of tags T, $\varrho: W \to [0,1]$ is a fuzzy membership function of a word-tag in a set of tags $\hat{T}$. The input of the model are a dictionary D, a set of words W in a tagged text, and a set of existing tags T. The output of the model is a set of tags $\hat{T}$, which is defined using the following TS fuzzy rule:

IF $\lambda(w)$ IS "high" AND $\mu(w)$ IS "low" AND $\nu(w)$ IS "high"

THEN $\hat{T} = \sigma(W), \forall w_i \in \hat{T} \subseteq W, \nexists w_j \in W, \varrho(w_i) < \varrho(w_j), w_i \neq w_j$ and $|\hat{T}| = 1 + \log_{10}|W|$.

We provide the following heuristic definition of the 3-relational operator $\varrho$ as follows:

$\varrho(w) = \lambda(w) \cap \overline{\mu(w)} \cap \nu(w)$,

here $\lambda(w) = f(w, W), w \in W$, here the function $\lambda(w)$ is equal to the frequency f of word w in the analysed text; $\mu(w) = \begin{cases} f(w, D), w \in D \\ 0, w \notin D \end{cases}$, here the function $\mu(w)$ is equal to 1 if the word w is not found in the dictionary D either because it is a rare word, or a new word (e.g., abbreviation) introduced by the user himself; otherwise it is equal to the frequency f of a word w in a text corpus represented by the dictionary D; $\nu(w) = \begin{cases} f(w, T), w \in T \\ 1, w \notin T \end{cases}$, here the function $\varrho(w)$ is equal to 1 if the tag-word w is not found in a set of tags T; otherwise it is equal to the frequency f of a tag-word w in T; $\overline{(\cdot)}$ is the fuzzy complement operator, and $\cap$ is the fuzzy intersection operator.

Following Wang [39], the tag fuzzy membership function is implemented using the interpretations of the fuzzy union and fuzzy complement operators as follows:

Algebraic: $\varrho(w) = \lambda(w) \cdot (1 - \mu(w)) \cdot \nu(w)$

Zadeh: $\varrho(w) = \min[\lambda(w), 1 - \mu(w), \nu(w)]$

Bounded: $\varrho(w) = \max[0, \lambda(w) - \mu(w) + \nu(w) - 1]$

Dubois: $\varrho(w) = \frac{\lambda(w) \cdot (1 - \mu(w)) \cdot \nu(w)}{\max[\lambda(w), 1 - \mu(w), \nu(w)]}$

Yager: $\varrho(w) = 1 - \min\left[1, \sqrt{(1 - \lambda(w))^2 + \mu^2(w) + (1 - \nu(w))^2}\right]$

Schweizer: $\varrho(w) = \frac{1}{\sqrt{\frac{1}{\lambda^2(w)} + \frac{1}{(1 - \mu(w))^2} + \frac{1}{\nu^2(w)} - 2}}$

Einstein: $\varrho(w) = \frac{\lambda(w) \cdot (1 - \mu(w)) \cdot \nu(w)}{1 + (1 - \lambda(w)) \cdot \mu(w) \cdot (1 - \nu(w))}$

Hamacher: $\varrho(w) = \frac{\lambda(w) \cdot (1 - \mu(w)) \cdot \nu(w)}{\lambda(w) + (1 - \mu(w)) + \nu(w) - \lambda(w) \cdot (1 - \mu(w)) - \lambda(w) \cdot \nu(w) - (1 - \mu(w)) \cdot \nu(w) + \lambda(w) \cdot (1 - \mu(w)) \cdot \nu(w)}$

Frank: $\varrho(w) = \log_2\left[1 + (2^{\lambda(w)} - 1)(2^{1 - \mu(w)} - 1)(2^{\nu(w)} - 1)\right]$

The Model-F2 fuzzy tagging algorithm is as follows:

1. Parse text into words.
2. Calculate fuzzy word membership in the text.
3. Calculate fuzzy word-tag membership using word-dictionary and tag-tagset fuzzy membership.
4. Select top word-tags.
5. Update tagset.

### E. Model F-3: Text, Dictionary,Tagset and Ontology

Model-F3 extends model-F2 with ontology (actually, a particular kind of ontology consisting only of concepts and their hypernymy/hyponymy relations).

Let the fuzzy-logic based tagging model M be a tuple $M = (L, D, W, O, T, \widehat{T}, \mu, \lambda, \nu, \chi, \xi, \varrho)$, where L is a language over alphabet $\Sigma$, D is a finite set of common words (dictionary) in a language L, $D \subset \Sigma^*$, W is a finite set of words in a text written in language L, $W \subset D$, T is a finite set of tags such that $T \subset D$, $|T| \ll |D|$, $\widehat{T}$ is a finite set of tags extracted from W such that $\widehat{T} \subset W$, $|\widehat{T}| \ll |W|$, $O = (C, R)$ is an ontology of concepts C and their relations R such that $C \subset D$, $\mu: D \to [0,1]$ is a fuzzy membership function of a word in a dictionary D which represents the frequency of its occurrence in a text corpus of a language L, $\lambda: W \to [0,1]$ is a fuzzy membership function of a word which represents its frequency in a text W, $\nu: T \to [0,1]$ is a fuzzy membership function of a tag which represents its frequency in a set of tags T, $\chi: C \to C$ is a hypernymyfunction that returns a hypernym of a concept from ontology O if it exists, $\xi: W \to T$ is the tag selection operator, and $\varrho: W \to [0,1]$ is a fuzzy membership function of a tag in a set of tags $\widehat{T}$. The input of the model are a dictionary D, a set of words W in analyzed text, an ontology O and a set of known tags T. The output of the model is a set of tags $\widehat{T}$, which is defined using the following TS fuzzy rule:

IF $\lambda(w)$ IS "high" AND $\mu(w)$ IS "low" AND $\nu(w)$ IS "high"

THEN $\widehat{T} = \sigma(W), \forall w_i \in \widehat{T} \subseteq W, \nexists w_j \in W,\ \varrho(w_i) < \varrho(w_j), w_i \neq w_j$ and $|\widehat{T}| = 1 + \log_{10}|W|$

We provide the following heuristic definition of the relational algebra selection operator $\varrho$ as follows:

$$\varrho(w) = \lambda(w) \cap \overline{\mu(w)} \cap \nu(w),$$

here $\lambda(w) = f(w, W), w \in W$, is the frequency f of a word w in the analyzed text; $\mu(w) = \begin{cases} f(w, D), w \in D \\ 0, w \notin D \end{cases}$, the function $\mu(w)$ is equal to 1 if the word w is not found in the dictionary D either because it is a rare word, or a new word (e.g., abbreviation) introduced by the user himself; otherwise it is equal to the frequency f of a word w in a text corpus represented by the dictionary D; $\nu(w) = \begin{cases} f(w, T), w \in T \\ 1, w \notin T \end{cases}$, the function $\varrho(w)$ is equal to 1 if the tag-word w is not found in a set of tags T; otherwise it is equal to the frequency f of a tag-word w in T; $\overline{(\cdot)}$ is the fuzzy complement operator, and $\cap$ is the fuzzy intersection operator.

We have implemented the following tag fuzzy membership function using the interpretations of the fuzzy union and fuzzy complement operators [39] as follows:

Algebraic: $\varrho(w) = \lambda(w) \cdot (1 - \mu(w)) \cdot \nu(w)$

Zadeh: $\varrho(w) = \min[\lambda(w), 1 - \mu(w), \nu(w)]$

Bounded: $\varrho(w) = \max[0, \lambda(w) - \mu(w) + \nu(w) - 1]$

Dubois: $\varrho(w) = \frac{\lambda(w) \cdot (1-\mu(w)) \cdot \nu(w)}{\max[\lambda(w), 1-\mu(w), \nu(w)]}$

Yager: $\varrho(w) = 1 - \min\left[1, \sqrt{(1-\lambda(w))^2 + \mu^2(w) + (1-\nu(w))^2}\right]$

Schweizer: $\varrho(w) = \frac{1}{\sqrt{\frac{1}{\lambda^2(w)} + \frac{1}{(1-\mu(w))^2} + \frac{1}{\nu^2(w)} - 2}}$

Einstein: $\varrho(w) = \frac{\lambda(w) \cdot (1-\mu(w)) \cdot \nu(w)}{1 + (1-\lambda(w)) \cdot \mu(w) \cdot (1-\nu(w))}$

Hamacher: $\varrho(w) = \frac{\lambda(w) \cdot (1-\mu(w)) \cdot \nu(w)}{\lambda(w) + (1-\mu(w)) + \nu(w) - \lambda(w) \cdot (1-\mu(w)) - \lambda(w) \cdot \nu(w) - (1-\mu(w)) \cdot \nu(w) + \lambda(w) \cdot (1-\mu(w)) \cdot \nu(w}$

Frank: $\varrho(w) = \log_2[1 + (2^{\lambda(w)} - 1)(2^{1-\mu(w)} - 1)(2^{\nu(w)} - 1)]$

The tag selection operator $\xi$ is defined as follows:

$$\xi(w_i) = \begin{cases} w_i, \text{if } \varrho(w_i) \geq \cup\varrho(w_j), \text{such as } \chi(w_i) = \chi(w_j) \\ \chi(w_i), \text{if } \varrho(w_i) < \cup\varrho(w_j) \end{cases},$$

here $\cup$ is the fuzzy n-ary union operator, which can be interpreted (following [37]) as follows:

Algebraic: $\cup\varrho(w_j) = 1 - \prod_j (1 - w_j)$

Zadeh: $\cup\varrho(w_j) = \max_j \{w_j\}$

Bounded sum: $\cup\varrho(w_j) = \min[1, \sum_j w_j]$

Yager: $\cup\varrho(w_j) = \min\left[1, \sqrt{\sum_j w_j^2}\right]$

Frank: $\cup\varrho(w_j) = 1 - \log_2\left[1 + \prod_j \left(2^{(1-w_j)} - 1\right)\right]$

The Model-F3 fuzzy tagging algorithm is as follows:

1. Parse text into words.
2. Calculate fuzzy word membership in the text.
3. Calculate fuzzy word-tag membership using word-dictionary and tag-tagset fuzzy membership.
4. Select top word-tags.
5. Find hypernym-tags of top word-tags from ontology.
6. Calculate fuzzy membership of hypernym-tags.
7. Select tags from word-tags and hypernym-tags.
8. Update tagset.

The algorithm will select tags only from the pre-prepared tagset, so high accuracy of the algorithm depends on the quality of a given tagset.

## IV. Case Study

The experiment was performed to establish the efficiency of web document tagging algorithms. We have used data aggregation sites (Delicious, BlogFlux, Technorati) to compile a collection of online texts divided into five categories: technology, nature, careers, science, cooking (20 texts in each category). Based on the length of texts we have divided all texts of each category into three groups: short texts (less than 150 words), medium length texts (from 150 to 600 words), and long texts (more than 600 words).

See the characteristics of the text dataset (given in [42] with some preliminary results) in Table I.

TABLE I. Average length of texts (in words) by category

| Category/length | Short | Medium | Long |
|---|---|---|---|
| **Technology** | 95 | 404 | 816 |
| **Career** | 87 | 418 | 1253 |
| **Cooking** | 174 | 638 | 1477 |
| **Nature** | 131 | 392 | 1159 |
| **Science** | 133 | 574 | 949 |

Each text was analyzed using four different algorithms (TF, TF-IDF, co-occurrence (COOC) [26] – as baseline methods, and the proposed Model-F1), as well as two different ways to choose the number of keywords as tags: fixed (TOP-5: top 5 keywords are identified and recommended as tags) and variable (LOG: the number of tags depend upon the length of the text following the logarithmic relationship $n = \log(N) + 1$, where $N$ is the number of words in the analyzed text, and $n$ is the number of recommended tags. As the texts selected from data aggregation websites already have the tags assigned by their authors, we have evaluated the accuracy of the tagging methods using the standard metrics of precision, recall and F-score (F). Precision shows that the recommended tags are correct, while recall shows that the algorithm can find correct tags. F-score metric combines both precision and recall.

## V. Results and Evaluation

The results of tagging are presented in Tables II-XVI.

TABLE II. Tagging accuracy results (short technology texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.00 | 0.14 | 0.15 | TF | 0.20 | 0.11 | 0.14 |
| IDF | 0.16 | 0.14 | 0.15 | IDF | 0.27 | 0.14 | 0.18 |
| COOC | 0.12 | 0.07 | 0.07 | COOC | 0.20 | 0.07 | 0.09 |
| F1 | 0.12 | 0.08 | 0.08 | F1 | 0.13 | 0.04 | 0.05 |

TABLE III. Tagging accuracy results (medium technology texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.48 | 0.32 | 0.37 | TF | 0.80 | 0.32 | 0.44 |
| IDF | 0.44 | 0.30 | 0.35 | IDF | 0.53 | 0.23 | 0.32 |
| COOC | 0.36 | 0.22 | 0.26 | COOC | 0.33 | 0.14 | 0.19 |
| F1 | 0.36 | 0.24 | 0.28 | F1 | 0.40 | 0.16 | 0.23 |

TABLE IV. Tagging accuracy results (long technology texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.56 | 0.45 | 0.49 | TF | 0.62 | 0.36 | 0.44 |
| IDF | 0.32 | 0.24 | 0.27 | IDF | 0.50 | 0.26 | 0.34 |
| COOC | 0.32 | 0.24 | 0.27 | COOC | 0.33 | 0.19 | 0.23 |
| F1 | 0.28 | 0.24 | 0.25 | F1 | 0.38 | 0.24 | 0.29 |

TABLE V. Tagging accuracy results (short career texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.32 | 0.21 | 0.25 | TF | 0.35 | 0.16 | 0.21 |
| IDF | 0.20 | 0.15 | 0.17 | IDF | 0.27 | 0.14 | 0.19 |
| COOC | 0.36 | 0.23 | 0.28 | COOC | 0.40 | 0.20 | 0.26 |
| F1 | 0.16 | 0.12 | 0.14 | F1 | 0.15 | 0.09 | 0.11 |

TABLE VI. Tagging accuracy results (medium careertexts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.48 | 0.43 | 0.45 | TF | 0.57 | 0.34 | 0.42 |
| IDF | 0.36 | 0.32 | 0.34 | IDF | 0.38 | 0.23 | 0.28 |
| COOC | 0.48 | 0.44 | 0.45 | COOC | 0.48 | 0.28 | 0.36 |
| F1 | 0.40 | 0.37 | 0.38 | F1 | 0.63 | 0.37 | 0.46 |

TABLE VII. Tagging accuracy results (long career texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.36 | 0.48 | 0.39 | TF | 0.45 | 0.48 | 0.44 |
| IDF | 0.28 | 0.35 | 0.29 | IDF | 0.35 | 0.35 | 0.33 |
| COOC | 0.44 | 0.55 | 0.47 | COOC | 0.45 | 0.48 | 0.44 |
| F1 | 0.28 | 0.32 | 0.28 | F1 | 0.37 | 0.31 | 0.31 |

TABLE VIII. Tagging accuracy results (short cooking texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.44 | 0.45 | 0.44 | TF | 0.48 | 0.38 | 0.42 |
| IDF | 0.20 | 0.20 | 0.20 | IDF | 0.15 | 0.10 | 0.12 |
| COOC | 0.44 | 0.48 | 0.45 | COOC | 0.53 | 0.45 | 0.47 |
| F1 | 0.28 | 0.28 | 0.28 | F1 | 0.40 | 0.28 | 0.32 |

TABLE IX. Tagging accuracy results (medium cooking texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.60 | 0.40 | 0.46 | TF | 0.65 | 0.31 | 0.41 |
| IDF | 0.44 | 0.31 | 0.35 | IDF | 0.48 | 0.25 | 0.31 |
| COOC | 0.56 | 0.39 | 0.44 | COOC | 0.58 | 0.34 | 0.41 |
| F1 | 0.32 | 0.21 | 0.24 | F1 | 0.38 | 0.18 | 0.23 |

TABLE X. Tagging accuracy results (long cooking texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.68 | 0.51 | 0.58 | TF | 0.70 | 0.41 | 0.52 |
| IDF | 0.48 | 0.36 | 0.41 | IDF | 0.55 | 0.32 | 0.41 |
| COOC | 0.52 | 0.39 | 0.44 | COOC | 0.55 | 0.32 | 0.41 |
| F1 | 0.44 | 0.32 | 0.37 | F1 | 0.55 | 0.32 | 0.41 |

TABLE XI. Tagging accuracy results (short nature texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.40 | 0.46 | 0.42 | TF | 0.45 | 0.32 | 0.36 |
| IDF | 0.20 | 0.22 | 0.21 | IDF | 0.32 | 0.20 | 0.24 |
| COOC | 0.20 | 0.25 | 0.22 | COOC | 0.25 | 0.20 | 0.21 |
| F1 | 0.36 | 0.42 | 0.38 | F1 | 0.57 | 0.38 | 0.44 |

TABLE XII. Tagging accuracy results (medium nature texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.32 | 0.25 | 0.28 | TF | 0.42 | 0.22 | 0.29 |
| IDF | 0.24 | 0.21 | 0.22 | IDF | 0.32 | 0.18 | 0.22 |
| COOC | 0.28 | 0.22 | 0.25 | COOC | 0.42 | 0.22 | 0.29 |
| F1 | 0.24 | 0.19 | 0.21 | F1 | 0.30 | 0.17 | 0.21 |

TABLE XIII. Tagging accuracy results (long nature texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.40 | 0.32 | 0.35 | TF | 0.37 | 0.22 | 0.28 |
| IDF | 0.48 | 0.38 | 0.42 | IDF | 0.47 | 0.29 | 0.35 |
| COOC | 0.36 | 0.29 | 0.32 | COOC | 0.32 | 0.19 | 0.24 |
| F1 | 0.24 | 0.19 | 0.21 | F1 | 0.27 | 0.16 | 0.20 |

TABLE XIV. Tagging accuracy results (short science texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.28 | 0.26 | 0.27 | TF | 0.27 | 0.27 | 0.27 |
| IDF | 0.12 | 0.17 | 0.14 | IDF | 0.20 | 0.20 | 0.20 |
| COOC | 0.20 | 0.22 | 0.21 | COOC | 0.33 | 0.33 | 0.33 |
| F1 | 0.16 | 0.15 | 0.15 | F1 | 0.27 | 0.27 | 0.27 |

TABLE XV. Tagging accuracy results (medium science texts)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.40 | 0.39 | 0.39 | TF | 0.52 | 0.53 | 0.52 |
| IDF | 0.20 | 0.21 | 0.20 | IDF | 0.28 | 0.28 | 0.28 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| COOC | 0.24 | 0.22 | 0.23 | COOC | 0.35 | 0.35 | 0.35 |
| F1 | 0.20 | 0.23 | 0.21 | F1 | 0.28 | 0.30 | 0.29 |

TABLE XVI. TAGGING ACCURACY RESULTS (LONG SCIENCE TEXTS)

| Top-5 | Prec | Recall | F | Log | Prec | Recall | F |
|---|---|---|---|---|---|---|---|
| TF | 0.20 | 0.25 | 0.18 | TF | 0.27 | 0.28 | 0.27 |
| IDF | 0.20 | 0.25 | 0.18 | IDF | 0.22 | 0.23 | 0.22 |
| COOC | 0.12 | 0.13 | 0.09 | COOC | 0.17 | 0.17 | 0.17 |
| F1 | 0.32 | 0.40 | 0.35 | F1 | 0.43 | 0.47 | 0.45 |

The results for F-score are summarized graphically in Fig. 1 and Fig. 2 for fixed and variable tagging respectively. Unfortunately, due to the small size of text dataset used in our experiments, there is some variability in the results.

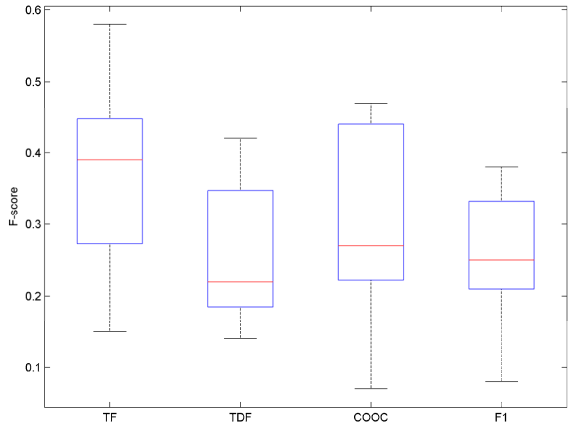


Fig. 1. Accuracy (F-score) of fixed tagging (Top-5)

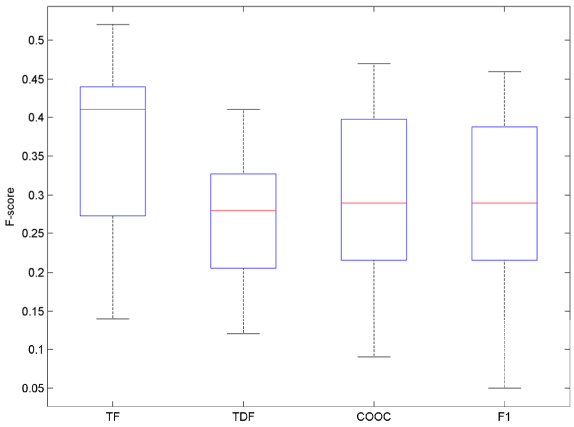


Fig. 2. Accuracy (F-score) of variable tagging (LOG)

To evaluate the difference of the tagging accuracy results of Model-F1 against each of the baseline methods (TF, IDF and COOC) statistically, we use the Wilcoxon signed-ranks test, which ranks the differences in performances of two methods for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences. The results of the test show that in case of Top-5 tagging, there was no significant difference between the accuracy (F-score) of F1 and IDF as well as F1 and COOC (at $\alpha$=0.05). In case of LOG tagging, (at $\alpha$=0.05), there was no significant difference between the accuracy (F-score) results reported by all methods. The results are summarized in Table XVII.

TABLE XVII. RESULTS OF WILCOXON SIGNED-RANKS TEST

| F1 vs TF | F1 vs IDF | F1 vs COOC |
|---|---|---|
| *Fixed number of tags (Top-5)* | | |
| $p = 0.0032$<br>h = 1 | $p = 0.6093$<br>$h = 0$ | $p = 0.0917$<br>$h = 0$ |
| *Variable number of tags (LOG)* | | |
| $p = 0.0176$<br>$h = 1$ | $p = 0.8199$<br>$h = 0$ | $p = 0.5310$<br>$h = 0$ |

The tagging results show that higher accuracy is obtained when analyzing longer texts. Better results can be obtained using variable number of recommended tags. The results of the proposed Model-F1 algorithm are better for short algorithms and are comparable with other known tag selection methods. However, for longer texts the proposed method performs relatively worse.

We expect to improve the results with the implementations of Model-F2 and Model-F3, which will use the personalized tagsets of users as well as ontological relations from WordNet to select better matching tags.

## VI. CONCLUSIONS AND FUTURE WORK

We have presented four fuzzy logic based models for automatic tag recommendation and with the experimental results of a fuzzy logic based model utilizing the users' dictionary information using texts of different topic and length. Results were compared with other known word frequency and co-occurrence probability based tag ranking methods.

The proposed text processing model has very important features that make it flexible for various texts processing tasks such as adapting the recommended tags to the existing set of user tags, creating a feedback for user, updating the set of tags and the dictionary after each tagging, supporting the personalization of tags, semantics-awareness by using an ontology to refine the recommended tagset.

The proposed model has the following characteristics as formulated in [43] for tag recommendation systems:

1) Generality: the tag recommendation system can automatically adapt to specific characteristics of different topics (technology, science, mature, cooking, career) as demonstrated by the performed experiments.

2) Adaptability: the recommendation can benefit from the already known tags used by the users thus ensuring feedback loop and adapting to current interests of a user.

3) Efficiency: the tag recommender uses simple fuzzy logic to handle online texts of common size in real time.

For the future work we plan to extend proposed approach to fit wider variety of texts by using personal tagset and ontology information, what can make the solution adaptive according to user preferences.

## ACKNOWLEDGEMENT

Authors acknowledge contribution to this project of Operational Programme: Knowledge, Education, Development financed by the European Social Fund under grant application POWR.03.03.00-00-P001/15.

## REFERENCES


[1] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", Scientific American Magazine, May 2001.

[2] M. Janik, A. Scherp, and S. Staab, "The Semantic Web: Collective Intelligence on the Web," Informatik Spektrum 34(5), p. 469-483, 2011.

[3] D. Brabham, "Crowdsourcing as a Model for Problem Solving: An Introduction and Cases", Convergence: The International Journal of Research into New Media Technological Studies 14(1), pp. 75–90, 2008.

[4] M. Gupta, R. Li, Z. Yin, and J. Han, "Survey on social tagging techniques," SIGKDD Explor. Newsl. 12, 1, pp. 58-72, 2010.

[5] L. Specia and E. Motta, "Integrating Folksonomies with the Semantic Web," Proc. of the 4th European conference on The Semantic Web: Research and Applications (ESWC '07). pp. 624-639, 2007.

[6] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su, "Optimizing web search using social annotations", Proc. of the 16th International conference on World Wide Web, ACM, pp. 501-510, 2007.

[7] C. Treude and M.-A. Storey, "How tagging helps bridge the gap between social and technical aspects in software development," in ICSE '09: 31st Int. Conference on Software Engineering, pp. 12–22, 2009.

[8] W. Dong and W.-T. Fu, "Toward a cultural-sensitive image tagging interface," Proc. of the 15th international conference on Intelligent user interfaces (IUI '10). ACM, New York, NY, USA, pp. 313-316, 2010.

[9] E. Giannakidou, F. Kaklidou, Y. Kompatsiaris, and A. Vakali, "Harvesting Intelligence in Multimedia Social Tagging Systems, Emergent Web Intelligence", Springer Verlag, Series: Studies in Computational Intelligence, pp. 135-167, 2010.

[10] M. Bernstein, D. Tan, G. Smith, M. Czerwinski, andE. Horvitz, "Collabio: A Game for Annotating People within Social Networks," Proc. of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST), pp. 97-100, 2009.

[11] K.A.F. Mohamed, "The impact of metadata in web resources discovering", Online Information Review 30(2), pp. 155-167, 2006.

[12] J. Goncalves, S. Hosio, D. Ferreira, and V. Kostakos, "Game of words: tagging places through crowdsourcing on public displays," Proc. of Conference on Designing Interactive Systems (DIS'14), pp. 705-714, 2014. DOI=http://dx.doi.org/10.1145/2598510.2598514

[13] S.O.K. Lee and A.H.W. Chun, "Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid ANN semantic structures," Proc. of the 6th WSEAS International Conference on Applied Computer Science (ACOS'07), Vol. 6, pp. 88-93, 2007.

[14] S. Golder, and B. Huberman, "Usage Patterns of Collaborative Tagging Systems," Journal of Information Science, 32(2), pp. 198-208, 2006.

[15] M. Diligenti, M. Gori, and M. Maggini, "Learning to Tag Text from Rules and Examples," Artificial Intelligence Around Man and Beyond - XIIth International Conference of the Italian Association for Artificial Intelligence, AI*IA 2011, pp. 45-56, 2011.

[16] M. F. Porter, "An algorithm for suffix stripping," in K.S. Jones and P. Willett (Eds.), Readings in information retrieval, Morgan Kaufmann Publishers Inc., pp. 313-316, 1997.

[17] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF, " Journal of Documentation 60 (5), pp. 503–520, 2004. doi:10.1108/00220410410560582.

[18] K.K. Bun and M. Ishizuka, "Topic Extraction from News Archive Using TF*PDF Algorithm," Proc. of the 3rd International Conference on Web Information Systems Engineering (WISE '02), pp. 73-82, 2002.

[19] S.Baskar, C.Deisy, N.R. Kalaiarasi, "A novel term weighting scheme midf for text categorization," Journal of Engineering Science and Technology, Vol. 5, pp. 94-107, 2010.

[20] R. Jin, C. Falusos, and A.G. Hauptmann, "Meta-scoring: automatically evaluating term weighting schems in IR without precision-recall," Proc. of the 24th ACM International Conference on Research and Development in Information Retrieval (SIGIR'01), pp. 83-89, 2001.

[21] C. Buckley, A. Singhal, and M. Mitra, "New retrieval approaches using SMART," 4th Text Retrieval conference (TREC-4), pp. 25-48, 1996.

[22] R. Navigli, and P. Velardi, "Semantic Interpretation of Terminological Strings," Proc. of 6th International Conference on Terminology and Knowledge Engineering (TKE 2002), Springer. pp. 95-100, 2002.

[23] Y. Park, R.J. Byrd, and B.K. Boguraev, "Automatic glossary extraction: beyond terminology identification," Proc. of the 19th international conference on Computational linguistics, Vol. 1, pp. 1-7, 2002.

[24] L. Kozakov, Y. Park, T.-H. Fin, Y. Drissi, Y. N. Doganata, and T. Cofino, "Glossary extraction and knowledge in large organisations via semantic web technologies," Proc. of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (Semantic Web Challenge Track), 2004.

[25] M.S. Ali, M.P. Consens, G. Kazai, and M. Lalmas, "Structural relevance: a common basis for the evaluation of structured document retrieval," Proc. of the 17th ACM conference on Information and knowledge management (CIKM '08), pp. 1153-1162, 2008.

[26] Y. Matsuo, and M. Ishizuka, "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information," FLAIRS Conference 2003, pp. 392-396, 2003.

[27] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation. Weirdness Indexing for Logical Document Extrapolation and Retrieval (WILDER)," J. of Machine Learning Research 3, pp. 993–1022, 2003.

[28] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles, "Real-time automatic tag recommendation," Proc. of ACM Conference on Research and development in information retrieval SIGIR'08, pp. 515–522, 2008.

[29] H.-C. Yang, and C.-H. Lee, "Mining tag semantics for social tag recommendation," Proc.of IEEE International Conference on Granular Computing, GrC 2011, art. no. 6122694, pp. 761-766, 2011.

[30] F.S. Tsai, "A tag-topic model for blog mining, " Expert Syst. Appl. (ESWA), 38(5), pp. 5330-5335, 2011.

[31] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent dirichlet allocation for tag recommendation," Proc. the Third ACM Conference on Recommender Systems RecSys '09, pp. 61-68. ACM, 2009.

[32] E. Montanes, J.R. Quevedo, I. Diaz, and J. Ranilla, "Collaborative Tag Recommendation System based on Logistic Regression," Proc. of the ECML PKDD Discovery Challenge 2009 (DC09), pp. 173-188, 2009.

[33] J.C. Platt and J.C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," Advances in Large Margin Classifiers, pp. 61–74. MIT Press, 1999.

[34] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," Advances in Neural Information Processing Systems 14, pp. 681–687. MIT Press, 2001.

[35] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," ACM Conference on Recommender Systems RecSys'08, pp. 43-50, 2008.

[36] S. Fujimura, K.O. Fujimura, and H. Okuda, "Blogosonomy: Autotagging Any Text Using Bloggers' Knowledge," Proc. of International Conference on Web Intelligence (WI'07). IEEE CS, pp. 205-212, 2007.

[37] J.M. Al-Kofahi, A. Tamrawi, T.T. Nguyen, H.A. Nguyen, and T.N. Nguyen, "Fuzzy set approach for automatic tagging in evolving software," Proc. of the 2010 IEEE International Conference on Software Maintenance (ICSM '10). IEEE CS, pp. 1-10, 2010.

[38] L.A. Zadeh, "Soft Computing and Fuzzy Logic," IEEE Software 11, 6, pp. 48-56, 1994. DOI=http://dx.doi.org/10.1109/52.329401

[39] H.F. Wang, "Comparative Studies on Fuzzy T-Norm Operators", Intl. J. of BUSFEL, 50, pp. 16-24, 1992.

[40] E.H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, 7(1), pp. 1-13, 1975.

[41] T. Takaki, and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst. Man Cyber. 15 pp. 116-132, 1985.

[42] R. Valys, "Research of web document automatic tagging methods", MSc. Thesis, Kaunas University of Technology, 2014.

[43] M. Lipczak and E. Milios, "Learning in efficient tag recommendation," Proc. of the fourth ACM conference on Recommender systems (RecSys '10). ACM, pp. 167-174, 2010.