FuzzyPhoto AHRC AH/J004367/1

Work Package 5 Report: Word Sense Disambiguation

Dr. Jethro Shell, Dr. Simon Coupland and Prof. Stephen Brown

August 21, 2013

Status: 2nd draft

Distribution Type: Internal

Keywords: Word Sense Disambiguation, Natural Language Processing, Part-of-Speech Tagging, Fuzzy Logic

Contents

1. Introduction

Incomplete and vague data are often common in art and humanities research, particularly in the data that is associated with historical artefacts such as photographs. Many collections suffer from inadequate documentation with researchers relying on the knowledge of curators to often identify relevant research materials. The increasing availability of resources through digital access and powerful search tools has opened up the opportunities to discover these resources, however the structure of the data can limit the accessibility.

The FuzzyPhoto project looks to investigate the use of Computational Intelligence (CI) techniques to identify relationships within the data through the production of a *finding aid* to suggest likely matches of photographs across different historical collections.

The FuzzyPhoto project aims to exploit information held within photographic metadata (data about the photographic data) contained within an AHRC funded database held by De Montfort University, the Exhibitions of the Royal Photographic Society 1870-1915 (ERPS) along side a number of partner databases. Photographic records provide a unique problem as they can be exhibited/published multiple times on different occasions using different titles by different people. Assigning a title to a specific image, as a result, can be complex.

In comparing metadata of the photographs held, a semantic similarity measure is used. A component of the similarity process is the use of query expansion. Query expansion is the process of using similar meanings to those in a query to increase the chances of matching words within the query itself [Xu et al. (1996)]. To undertake query expansion, an understanding of a word is required. The context surrounding individual words can influence the meaning. Understanding a word in its context is seemingly easy for the humans but a complex and difficult task for computers, the word *fast* can mean *rapid* but also *motionless* dependent on the context. To understand the context of words in a sentence, Part-Of-Speech Tagging (POST) can be used. POST is the application of descriptors for each element in a sentence to help disambiguate the words, for example the structure of the tagging output can be represented as:

The/DT Eiffel/NNP Tower/NNP looked/VBD beautiful/JJ in/IN the/DT sun/NN

where DT is a determinant, NNP is a proper noun (singular), JJ is an adjective, IN is a preposition, VBD is a verb (in the past tense) and NN is a noun (singular).

This report details the research carried out into the application of POST to assist Word Sense Disambiguation (WSD). POST was applied to photographic titles within the ERPS database with the aim of achieving an automated tagging process. The automated disambiguation of titles was highlighted as a required component in the FuzzyPhoto record matching process. To achieve this, a proposed method combining the use of fuzzy logic and probability was investigated. This was compared to two available software Part-of-Speech taggers. A comparison was made based upon accuracy. It was found that the fuzzy approach was overall less accurate than the compared software when used against a test dataset. Tagging 100 titles from the ERPS database consisting of 465 words, the fuzzy methodology achieved 83.65% accuracy. The Natural Language Toolkit (NLTK) tagger, in comparison, achieved 85.80%, with the Stanford software producing 86.23%. The fuzzy approach was more accurate at identifying particular word types within the structure of the sentence than the other methods compared. Overall, a recommendation was made to use the Stanford tagger within this project.

Outputs of this workpackage comprise:

1. This report.
2. Part-Of-Speech tagged annotated records of the ERPS database.
3. A Java implementation of Fuzzy Part-Of-Speech Tagging.

The elapsed time for this work package was 7 months. The resources required to complete this work package were 84 person-days.

2. Word sense disambiguation

Word Sense Disambiguation (WSD) is an open problem within computational linguistics. Understanding the meaning of a word in its context is something that is familiar and seemingly easy for a human, yet complex to a computer. For example, the word light can mean not heavy or illumination. The sentence *He turned on the light* is clear as to the meaning although the process to decide this is complex. Stevenson and Wilks defined WSD as

> "the process of identifying the meaning of words in context [Stevenson and Wilks (2003)] "

The FuzzyPhoto project uses elements of meta-data that are descriptive. They contain sentence structure that require contextual understanding. One of the steps within the FuzzyPhoto approach is the use of query expansion. Query expansion is the process of using similar meanings to those in a query to increase the chances of matching words within the query itself [Xu et al. (1996)]. To undertake query expansion, an understanding of a word is required. This is referred to as disambiguation.

There are a number of strategies that can be employed to disambiguate words within a context. Many focus on the use of a corpus taken from a specific subject area. The FuzzyPhoto project is different in this manner. There is not a single subject that encompasses the collections, as with medical documents or items from a specific location. Additionally the structure of the titles that are being focussed on are extremely sparse. Standard disambiguation approaches use text structures that encompass long sentences, or even paragraphs. This allows a deeper understanding of the context. The photographic metdata holds short text, the title averaging only 6.1 words. This produces a more difficult problem domain.

One element of WSD is the use of Part-Of-Speech Tagging (POST). POST is the process of assigning descriptors, or tags to input tokens within a sentence structure. The application of word classes has been used within linguistics since c.100 BC when Thrax used classification words such as noun, verb, participle, article, pronoun, preposition, adverb, and conjunction [Voutilainen (2003)]. POST is predominantly used as a preprocessing measure. It can be carried out quickly and more accurately than parsing, and development of taggers for specific domains is more rapid [Brill (2000)] . Specific POST systems have been focussed on individual problem domains such as Twitter [Gimpel et al. (2010)] and across languages [Snyder et al. (2009)].

The output from a POST process can be used to further disambiguate a sentence structure or be directly used as the sole disambiguation. The structure of the tagging output can be represented as:

The/DT Eiffel/NNP Tower/NNP looked/VBD beautiful/JJ in/IN the/DT sun/NN

where DT is a determinant, NNP is a proper noun (singular), JJ is an adjective, IN is a preposition, VBD is a verb (in the past tense) and NN is a noun (singular).

In this report, the use of a combined Fuzzy Logic (FL) and probabilistic method will be discussed. The focus of the method will be the extraction of word tags for 100 test titles from the Exhibitions of the Royal Photographic Society (1870-1915) (ERPS) database. The method will be compared to two developed, open source software applications based upon the accuracy of the tagging process.

3. Comparative Work

There are a large number of systems that have implemented POST. In this section a brief summary will be given of some of the approaches and current systems that are available.

3.1 Bidirectional Processing

Many processes approach the problem of POST t_1 as a unidirectional sequence problem. Despite the application of different algorithms, the system will navigate through the sequence in a single direction, left to right or right to left. In a standard left to right first-order Hidden Markov Model (HMM), the current tag t_0 is predicted based on the nature of the preceding tag,[Toutanova et al. (2003)]. Despite unidirectional models ability to capture the information of both directions of the model, as this is implicit when the next word is generated t_{-1} , bidirectional models have been proposed. Toutanova et al. [Toutanova et al. (2003)] exploit the use of dependency networks to efficiently infer more information from the sequence using both directions. Additionally they incorporated multiword feature templates so that idiomatic word sequences could be learnt and used within the model. Based on the model, Toutanova et al. were able to produce an accuracy of 97.24% per tag and 56.34% on correct whole sentence identification for the Penn Treebank WSJ dataset [Marcus et al. (1993)].

The work of Toutanova et al. [Toutanova et al. (2003)] forms the basis for the Stanford log-linear part-of-speech tagger [11]. The software implementation will be used as a comparison to the methodology in this report.

3.2 Support Vector Machine

Within classification problems, the use of a Support Vector Machine (SVM) is a popular approach. Giménez and Màrquez [Giménez and Màrquez (2004)] discuss the use of a SVM for POST. A SVM is a process to classify data by maximising the solution between two groups. Figure 1 shows the optimal hyperplane that is produced between two groups by maximising the margin at the closest points. The SVM is predominantly used to form a model of the POST domain based upon training data. The model can then be applied to the classification problem. Giménez and Màrquez used a SVM approach to tag 18 sections of the Wall Street Journal corpus. 2.81% of the words in the dataset were unknown to the model. They were able to generate tags with between 96.89% and

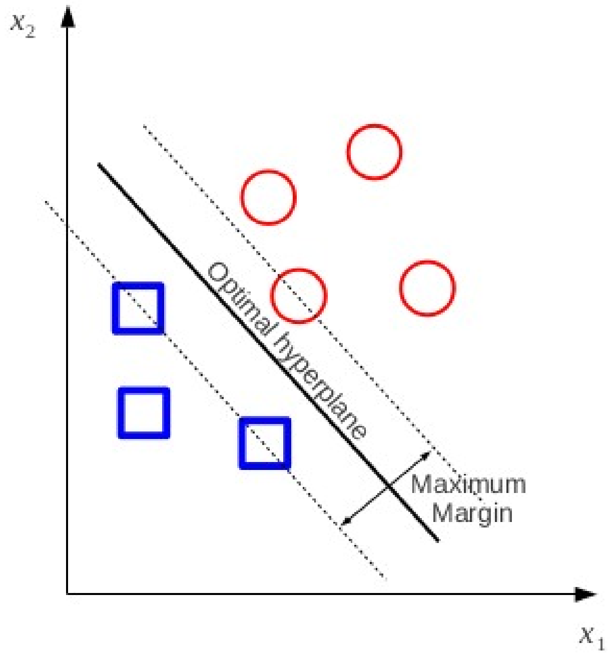

Figure 1. Outline of Support Vector Machine

98.96% accuracy.

3.3 Hidden Markov Model

There are a number of implementations that use a HMM [Manning (2011)], [Gimpel et al. (2010)]. In principle a HMM is a tool for representing probability distributions over a sequence of observations. A HMM is highly applicable to POST as a constituent part of the Markov process is to encapsulate all of the history of a process to predict the future of that process. Cutting et al. [Cutting et al. (1992)] used a HMM method to approach POST. Their strategy was to initially generate a HMM based on annotated sentences. The model was tuned using empirical and *a priori* information. The model was trained on approx. 500,000 words from the Brown corpus [Marcus et al. (1993)]. When applied to an equal quantity, a 96% accuracy was gained.

4. FuzzyPhoto Approach

This report proposes a new strategy for POST using a combination of probability and fuzzy sets to represent the uncertainty when defining a word type. This can be categorised as fuzzistics, the merging of the words fuzzy and statistics. Originally used in [Mendel (2007)], fuzzistics describes the problem of going from word data collected from a group of subjects quantified by statistics, with the uncertainties that are inherent, to a word fuzzy set model that captures the uncertainties within the word data [Gimpel et al. (2010)]. The outlined process uses a group of probabilistic methods to produce a group of differing values relating to each possible word type for each word within the title. These groups are

captured within type-1 fuzzy sets. The application proposed incorporates three sources of information. The sources of data come from the use of the Stanford parser [Toutanova et al. (2000)], an annotated corpus using data from the ERPS database and data captured from the WordNet lexical database [Miller and others (1995)].

The methodology is composed of four distinct stages:

1. Language Recognition.
2. Word Type Probability.
3. Fuzzification
4. Set Comparison

Each of these stages will be addressed in sequence using a simple example. To demonstrate the application of the approach, the use of a two word title will be used: **daffodil fair**.

4.1 Stage One: Language Recognition

The overall probabilistic structure is based upon the composition of the individual probabilities that a word embodies a particular type. Each word is contextualised by the words that exist around it. The words that precede a word will influence the word type. The titles within the ERPS database contain multiple language types. To overcome this issue, two third-party processes were used to identify and translate non-English language words. The process used a combination of information from language detection software [Shuyo (2010)] and the Microsoft Bing web-based translator. A reduced set of languages were selected within the detection software. It was found that predominantly French, German, Spanish and Dutch were within the ERPS dataset.

4.2 Stage Two: Word Type Probability

The second stage of the process produces a series of probabilities relating word types to each word within the title. The POST word types used are mapped to those within a modified WordNet structure. These types were: adjective, adverb, article, conjunction, determiner, noun, other, preposition, pronoun, and verb. There are three separate information sources that provide probabilities of the word formats. Each of these will be taken in turn.

1. Parsed Title - Extraction of the probability of POST through the use of the Stanford tagging software.
2. Annotated Text - Application of sentence sequence structure based on an annotated corpus.
4. WordNet Frequency Count - Incorporation of word type frequency counts using the WordNet corpus.

4.2.1 Parsed Title

Each of the titles contained within the corpus are processed through the Stanford Parser [Toutanova et al. (2000)]. The parser software is available as a downloadable library (see http://nlp.stanford.edu/software/lex-parser.shtml) that can be implemented through the use of the Java programming language. Additionally there are command line options and currently an online version. The simplest approach to produce an output is to use the command line operation. There are a number of options built into the parser that can be refined. A sentence can be parsed using:

```
java -mx200m -cp "stanford-parser.jar:." ParserDemo2
englishPCFG.ser.gz testsent.txt
```

where testsent.txt is the text file. The sentence is passed through the parser as a whole. The parser is configured to supply a normalised probabilistic value of the tag for each word based on the whole sentence. Taking our toy example daffodil fair, the parser returns a value for daffodil of 0.8141 for the tag noun. In the Fuzzy approach, a Java based implementation was used. The library was configured to incorporate the standard englishPCFG training structure.

4.2.2 Annotated Text

To supplement the information gained from the Stanford parser, an annotated corpus based on the ERPS database was used. Each record was taken and tagged by members of the FuzzyPhoto team experienced in data analysis. Labelled text is predominantly used to train systems to directly tag items. The sparsity of the available text coupled with the depth of subject area made this a difficult task. To gain information from the available data, the tag structure was extracted from 100 labelled titles taken from the ERPS database. This was based on a left-to-right sequence with a bi-gram structure. A probability of a second word type occurring was produced based upon the preceding word type. At the start of the sentence, the first word would be defined without the use of this process.

Taking the previously defined example, Daffodil is defined as being a noun. Based upon the accrued bi-gram information, the following word is assigned a set of tags mapped to probabilities. The top three probability values can be seen in Table 1.

W_1	Tag_1	W_2	Tag_2	Probability
daffodil	noun	fair	article	0.0283
daffodil	noun	fair	noun	0.2291
daffodil	noun	fair	adjective	0.0283

Table 1: Bigram of Word Sequence

4.2.3 WordNet Frequency Count

The third stage uses the WordNet software structure [Fellbaum (1998)]. At the heart of WordNet is an annotated corpus that contains over 117,000 synsets of nouns, verbs, adjectives and adverbs. WordNet contains a frequency of the occurrence of each annotated word within each synset group. Based upon this frequency, a probability of each word within the sentence structure is generated.

Using the Daffodil Fair example, WordNet would output the below probabilities:

$$daffodil \rightarrow noun \rightarrow 1.4634^{-4}$$
$$fair \rightarrow adjective \rightarrow 0.0196$$
$$fair \rightarrow noun \rightarrow 1.4634^{-4}$$
$$fair \rightarrow verb \rightarrow 9.309210^{-5}$$

Based on the frequency values, the tags noun and adjective would be used. Other values produced an output of 0 as no frequencies were defined.

4.3 Stage Three: Fuzzification

The values from each of the three component groups are fuzzified into a set representation. This uses the concepts set out by Lotfi Zadeh [Zadeh (1988),Zadeh (1965) of fuzzy logic. Fuzzy logic is an approach to representing degrees of truth rather than the classical view of logic as only true or false. A fuzzy set can be used to assign an element a degree of membership rather than simply a boolean value. This can represent *absolutely true* and *absolutely* false, but equally any degree in between. This assists in representing cases that are vague such as propositions like *this person is old.* Fuzzy sets are constructed in a domain that represents the subject matter. In the case of the implementation demonstrated here, the sets are formed based on the domains from each information group, and from each word type. Each set, constructed as a triangular set here though this can be a number of different functions, represents the linguistic values of Very Small (VS), Small (S), Medium (M), Large (L) and Very Large (VL). The sets illustrate the probability that the word *daffodil* is a noun. Figure 1 shows that *daffodil* produces a membership value in the set of Large of 0.8. It also produces a membership value of 0.2 in the set of Very Large.

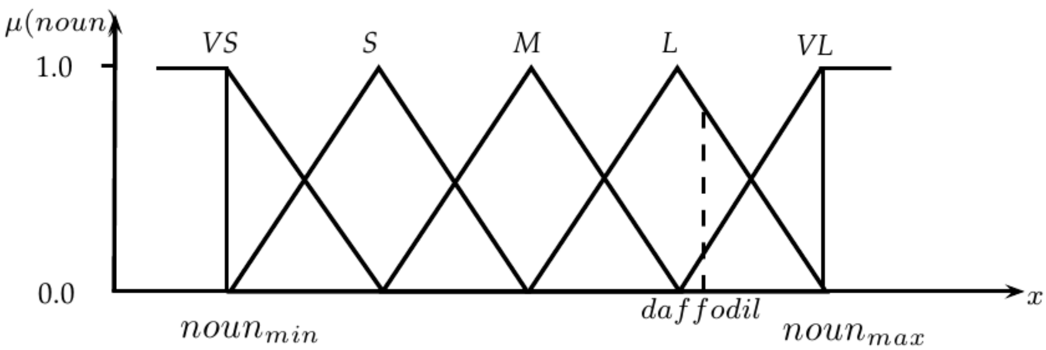

Figure 2. Fuzzy Representation of the Probability of the Word *daffodil* Produced By the Stanford Parser.

Once each of the component probabilistic elements are represented as fuzzy sets, each value is combined as a Fuzzy Weighted Average (FWA). The FWA was first proposed by Dong and Wong [Dong and Wong (1987)]. There have been a number of variations based on this original version [Lee and Park (1997),Liou and Wang (1992),Wu and Mendel (2010)]. This implementation uses the original structure. A FWA can be depicted using the following definition. If $A_{1,} A_{2,} \ldots, A_n$ and $W_{1,} W_{2,} \ldots W_n$ are fuzzy numbers defined in the universes $X_{1,} X_{2,} \ldots, X_n$ and $Z_{1,} Z_{2,} \ldots, Z_n$ then a fuzzy weighted average can be:

$$y = f\left(x_n, w_n\right) = \frac{x_1 w_1 + x_2 w_2 + \ldots x_n w_n}{w_1 + w_2 + \ldots w_n}$$

where for each $i=1,2,\ldots,n,$ $x_1 \in X_i, w_i \in Z_i,$ and $w_1 + w_2 + \ldots + w_n > 0$ .The use of a weighted average allows for the methodology to exert differing influence across the component elements within the system. Fuzzy sets were constructed based upon knowledge elicited from the data by the FuzzyPhoto team. The sets consisted of input values of each of the three components: Parsed Title, Annotated Text, WordNet Frequency Count. The annotated text was given the lowest weighting. Despite being data from the same context, the relatively small quantity of information was found to skew the test data. Larger weightings were given to the parsed and WordNet information.

4.4 Stage Four: Set Comparison

To compare each of the component sets within the method, a simple set comparison method was employed. The output from the FWA is a fuzzy set. A fuzzy set is produced for each element within each of the domains. For example, the annotated data element will produce a fuzzy set for each of the word types adjective, adverb, article, conjunction, determiner, noun, other, preposition, pronoun, and verb. These are defuzzified to produce a single value using the Centre of Gravity (CoG) [Zadeh (1994)] approach. Each of the values are compared. The highest value from each of the types

classifies the
corresponding word. Based on the example, the fuzzy method outputs daffodil as a noun, and fair also as a noun.

5. Experimentation

To test the methodology, a dataset of 100 titles from the ERPS database were additionally annotated by members of the FuzzyPhoto team along with the 100 titles used to build the model. The 100 annotated titles acted as ground truth. The methodology was applied to these values to assess its performance.

A comparison was also made against two readily available software implementations, the Stanford Part of Speech Tagger [Toutanova et al. (2000)], and the Natural Language Toolkit (NLTK) [Bird (2006)] Python software library.

An example of the format of the output from the fuzzy approach is shown in Table 2

Title No	Word	Type	Word Annotated	Type Annotated
3	triptographic	adjective	Triptographic	noun
3	cameos	noun	Cameos	noun
4	portrait	noun	Portrait	noun
4	of	preposition	of	preposition
4	a	determiner	a	article
4	lady	noun	Lady	noun
5	the	determiner	The	article
5	village	noun	Village	noun
5	wholesale	adjective	Mayor	noun
6	the	determiner	The	article
6	old	adjective	Old	adjective
6	church	noun	Church,	noun
6	bonchurch	verb	Bonchurch	noun
7	cloudland	adjective	Cloudland	noun
7	sunset	noun	Sunset	noun
8	high	adjective	High	noun
8	rocks	noun	Rocks,	noun
8	cheddar	noun	Cheddar	noun
8	cliffs	noun	Cliffs	noun
9	changing	verb	Changing	verb
9	box	noun	Box	noun
9	for	noun	for	preposition
9	dry	noun	Dry	verb
9	plates	noun	Plates,	noun
9	and	adverb	and	conjunction
9	expanding	adjective	Expanding	verb
9	camera	noun	Camera	noun

Table 2. Comparison of Fuzzy Method to Ground Truth Data.

In order to standardise the input, all of the text was reduced to lower case. A value of one was given to each correct tag, and zero to an incorrect tag. A standard error rate was produced from this process.

Running the fuzzy method across the annotated ground truth titles produced a word by word accuracy of 83.65% across the 100 titles. The titles were composed of 465 words in total. Table 3 shows a breakdown of the results by type.

Type	Total	Identified	Correctly Identified %
adjective	33	12	36.36
adverb	12	5	41.67
article	44	44	100
conjunction	17	17	100
noun	282	248	87.94
preposition	52	49	94.23
pronoun	2	2	100
verb	23	12	52.17

Table 3. Percentage of Correctly Identified Types For the Fuzzy Method.

The methodology was efficient at tagging articles, conjunctions, prepositions, pronouns and nouns. The system incorporated two basic expert constructed rules to assist the tagging. These facilitated the tagging of conjunction and preposition words. All “the” and “a” words were classified as prepositions. and words were defined as conjunctions. Further expansion of basic rules could bring further results. The quantity of pronouns within the test set was limited (only two). This limits the impact of this group. The system was less able to identify adverbs, adjectives and verbs. The system incorrectly identified a number of adjectives as nouns. This may result from the contextual nature of the title combined with the weighting structure. This requires further investigation.

To form a comparison, the test dataset was run against the NLTK implementation of a Part-of-Speech tagger. The implementation that was used was configured on the bi-gram tagger structure and trained using the Penn Bank annotated text. The NLTK software was able to achieve an 85.80% accuracy. The NLTK implementation outperformed the fuzzy method in the adverb (6), noun (263) and preposition (50) groups. It performed equally or less well in the other categories.

The test dataset was also processed using the Stanford tagger. The software is readily available and can be trained on any specific set of data. As no large set of annotated data was available, the tagger was trained using the supplied English language model. Overall the tagger was able to achieve an accuracy of 86.23%. It outperformed the fuzzy method within the adjective (14) and noun (258) categories, but as with the NLTK software it performed less or equally well in the other categories.

The comparison methods were more accurate in tagging the noun group. This was the largest group in the test dataset. The fuzzy method was able to outperform or match the NLTK implementation in four of the groups achieving 7% higher accuracy across those groups. In comparison to the Stanford methodology, the fuzzy method achieved equal or greater accuracy in five of the categories with a 0.02% increase.

6. Conclusion

The main conclusion of this report is to suggest that the Fuzzy Photo project adopts the use of the Stanford POST methodology. The combination of accuracy, when used on the ERPS test dataset, and speed of implementation are in tune with the developmental process.

However, there is scope to further investigate and expand the fuzzy disambiguation methodology. The use of fuzzy sets to represent the part-of-speech tags is able to assist in the representation of the uncertainty that is contained within text. The fuzzy method was close in performance to the two methods compared. Certain categories within the titles were shown to be outperformed by the fuzzy method in comparison to the NLTK and Stanford software.

Further work may yield more results. A greater corpus to extract the probabilistic values, and the use of bidirectional processing may improve the accuracy gained from the methodology. Additionally, the adoption of further rules within the system may refine the process.

References

Bird S. (2006) NLTK: the natural language toolkit. , 69-72.
Brill E. (2000) Part-of-speech tagging. *Handbook of Natural Language Processing* , 403-414.

Cutting D., Kupiec J., Pedersen J. and Sibun P. (1992) A practical part-of-speech tagger. , 133-140.
Dong W. and Wong F. (1987) Fuzzy weighted averages and implementation of the extension principle. *Fuzzy sets and systems* **21**, 183-199.
Fellbaum C. (1998) WordNet: An electronic database. , .
Giménez J. and Màrquez L. (2004) SVMTool: A general POS tagger generator based on Support Vector Machines. , .
Gimpel K., Schneider N., O'Connor B., Das D., Mills D., Eisenstein J., Heilman M., Yogatama D., Flanigan J. and Smith N. A. (2010) Part-of-speech tagging for twitter: Annotation, features, and experiments. , .
Lee D. H. and Park D. (1997) An efficient algorithm for fuzzy weighted average. *Fuzzy sets and systems* **87**, 39-45.
Liou T.-S. and Wang M.-J. J. (1992) Fuzzy weighted average: an improved algorithm. *Fuzzy sets and systems* **49**, 307-315.
Manning C. D. (2011) *Part-of-speech tagging from 97% to 100 is it time for some linguistics?*. In: (Ed.), *Computational Linguistics and Intelligent Text Processing*, Springer.
Marcus M. P., Marcinkiewicz M. A. and Santorini B. (1993) Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* **19**, 313-330.
Mendel J. M. (2007) Computing with words and its relationships with fuzzistics. *Information Sciences* **177**, 988-1006.
Miller G. A. and others (1995) WordNet: a lexical database for English. *Communications of the ACM* **38**, 39-41.
Shuyo N. (2010) Language Detection Library for Java. , .
Snyder B., Naseem T., Eisenstein J. and Barzilay R. (2009) Adding more languages improves unsupervised multilingual part-of-speech tagging: a Bayesian non-parametric approach. , 83-91.
Stevenson M. and Wilks Y. (2003) Word sense disambiguation. *The Oxford Handbook of Comp. Linguistics* , 249-265.
Toutanova K., Klein D., Manning C., Morgan W., Rafferty A., Galley M. and Bauer J. (2000) Stanford log-linear part-of-speech tagger. , .
Toutanova K., Klein D., Manning C. D. and Singer Y. (2003) Feature-rich part-of-speech tagging with a cyclic dependency network. , 173-180.
Voutilainen A. (2003) Part-of-speech tagging. *The Oxford handbook of computational linguistics* , 219-232.
Wu D. and Mendel J. M. (2010) Ordered fuzzy weighted averages and ordered linguistic weighted averages. , 1-7.
Xu, J., & Croft, W. B. (1996) Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 4-11). ACM.
Zadeh L. (1965) Fuzzy sets*. *Information and control* **8**, 338-353.
Zadeh L. (1988) Fuzzy logic. *Computer* **21**, 83-93.
Zadeh, L. (1994) Soft computing and fuzzy logic *Software, IEEE,* **1994**, *11*, 48 -56