Contents

1 Getting Started

1.1 Introduction

In the first chapter of our thesis we introduce you chiefly to the theory where this thesis is based. First and foremost, we guide you around the "sight-seeings" of Natural Language Processing by presenting you both science and fantasy, pointing out the basics of our thesis as well; moreover, we display the older related work (actually, our theory base), and, then, we point out our thesis-contribution. Have a nice reading!

1.2 NLP sight-seeings and thesis basics

A very interesting and quite challenging scientific field of computer engineering is Natural Language Processing (NLP). Many scientists, many years now, almost since the birth of computer era, cope with a variety of language-related research, e.g. automated translation, semantic similarity, spelling correction, grammar checking, information retrieval, speech recognition-understanding, speech synthesis, spoken dialogue systems, even lip-reading! Remember the film "A Space Odyssey", HAL 9000! The truth is that although these problems are far from completely solved, much of the language-related "HAL-technology" is currently being developed, with some of it already available!
Besides the clear-seen applications we mentioned above, there is plenty of other seemingly irrelevant, "indistinguishable" NLP-software. Consider Web applications. A characteristic paradigm could be this: Everyone knows Amazon. Imagine when you pick your new favorite book, background super-scripts running. Imagine these codes to scan your beloved book, in order to find key-words (probably with high frequency), and then, next page... you have not only the common: "customers who bought this item also bought bla-bla-bla", but super-suggestions as well! Yeah, I know... sounds cool, but let's get reorganized again. Not many lines were necessary to find out how truly interesting is the NLP area, and computer science in general. But the question still remains: what is *this* all about?
Well, nothing more nothing less, but a NLP-area: Word Semantics, and, namely, *Unsupervised Semantic Similarity Computation using Word Senses and Web Search Engines*.
First and foremost, by saying "semantic similarity computation", we mean, of

course, semantic similarity between two words. Furthermore, when we say "unsupervised" we mean "automated". Contemporary research makes titanic efforts in order to "set free mankind" from constructing, managing, supporting and maintaining huge and complex databases like WordNet. Moreover, "word senses" are, we could say, word-groups, parts of a "big" context, which help us to find similarity-connections between... contexts.
Concisely, this diploma thesis consists of two parts: one for Greek language, and one for English language. Both have common base: to achieve a high correlation in semantic similarity between humans and machines! And this, the whole procedure, as we said above, is being done totally unsupervised, i.e. we download (automatically, using Yahoo Search API), e.g., 100 contexts (from 100 URLs) for each of the two words, we grab their occurrences, and at last compare them by running metrics. All automated! And what have we achieved?
Well, definitely not Everest! The results are quite good, and set a solid base for further research on subject (especially for Greek language). As we said at the beginning, the NLP research area is, not only quite challenging (as most computer engineering fields), but also "open"! – And as a conclusion: *That's our try, that's our result, and that... will be your go by!*

1.3 Related work

We begin to unfold the previous related work – which, of course, makes the base of our research – by briefly presenting you the very important paper of *"Contextual Correlates of Semantic Similarity"* [2]. This paper investigates the relationship between semantic and contextual similarity for pairs of nouns that vary from high to low semantic similarity. The subjects of the research are students and they were told to classify the pairs by semantic similarity (beginning from the higher to the lower one) – at this point, we must say that the set of pairs made by choosing specific ones from a bigger set (see more in *"Contextual Correlates of Synonymy"* [1]), in order to correlate their results. Their conclusions were the elaboration-confirmation of a claim: "that several linguistics and psychologists believe there is a close connection between the meanings of words and contexts in which they use".
Another important paper is that of *"Unsupervised Semantic Similarity Computation using Web Search Engines"* [3]. The concept in this paper is same with that we mentioned above, i.e. similar words mean similar contexts. The set (of

pairs) is the same with that of Charles and Miller, and, similarly, there is a group of subjects (mainly students) that are called to classify the pairs, by semantic similarity. The crucial difference, in this paper, is that the same job (semantic similarity classification) does computer (unsupervised, i.e. downloading, automatically, internet contexts) as well, in order to achieve a high correlation between humans and computers. The results are very good, especially, when they are compared to these of supervised methods (look for more in [4]).
These are the older papers that we use as base of our research. All of them concern English language. Our try, is to implement them (theory and concepts) in Greek language, as also in English (in a different way). The difference between Greek and English language is that in English language we are going to implement "word senses", which means that we don't use the entire context, but parts of it. More simply, we cluster the big context and apply our "semantic-similarity metrics"[1] between, e.g., the two prevailed clusters[2]. These clusters called *senses*. At next paragraph, we describe the basic concepts and claims of the theory.

1.4 Theory

As far as theory is concerned, this is the basic *claim*: that there is a close connection (as several linguistics and psychologists believe) between the meanings of words and contexts in which they are used - and this claim we are trying to see how it works on both languages.
The idea is simple. Consider next pair: *automobile - car*, and consider two contexts, which contain the words correspondingly. Now, assume a Window Size (WS) which we apply on context. How?
Firstly, we find the keyword (automobile or car in our example) and then we grab its neighboring context, simply, by applying the WS (from both sides). In order to understand better this paradigm and in order to clarify every question , we "visualize" the following example:

[1] Look for them in next chapter.
[2] We use choice criteria, look for them on chapter 5.

For keyword "automobile" the neighboring context (consider a WS equal to 3) could be this[3]*:*

auto shows at *automobile* var new var
warranty subscribeswf tower *automobile* http static automobilemag
so not an *automobile* but we ve
an last of *automobile* born from years
window attachevent onload *automobile* magazine source interlink
into an early *automobile* passenger cars in
per people an *automobile* via french from
name for an *automobile* is a car
mechanical vehicle or *automobile* in about this
working three wheeled *automobile* this was at
in paris an *automobile* powered by an
of the modern *automobile* in benz was

For keyword "car" the neighboring context (also consider a WS equal to 3) could be this:

prices sell my *car* find auto function
length certified used *car* mercedes search more
value your zip *car* shopping tools and
suvs top and *car* best worst gas
more best worst *car* shopping tools and
hot dealsnegotiating with *car* photos and auto
recalls present more *car* shopping car talk
more car shopping *car* talk s maintenance
repair tipsfix your *car* find a mechanic
the blog for *car* blog at month
bmw more video *car* expert car nissan
video car expert *car* nissan sentrase of

After that, and since we have isolated the neighboring context, we apply context metrics[4] and we, finally, get the wanted semantic-similarity grade in order to calculate the correlation between humans and computers!

[3]Note that the data are real... derived from the internet

[4]Read about them in the next chapter. Concisely, they are metrics which use the keyword contexts in order to produce a semantic-similarity grade!

1.5 Theory contribution

Our contribution on theory is, at first, simply our try to implement it mainly in Greek language. Greek is a completely different language in syntax, grammar and structure in general. For that reason "Greek area", as also any other "non-English area", is very interesting and intriguing by research perspective in order to investigate how the metrics work, and how the language works grammatically, syntactically and generally in structure. Furthermore, in our contribution is also the *"unsupervised way"* we are trying to implement our research (both in Greek language with "big contexts", and in English language with "senses") appends as well. Algorithmic stages (that we present and analyze precisely in the next chapter) as html downloading and filtering, contexts manipulation, metrics implementation and correlation extraction are all done automatically and, particularly, the crucial automated stage of *pair-similarity extraction* is *independent* (since we use web search engines) of any kind word or context databases, which (databases) need renewal and appropriate feedback, maintenance and support!

1.6 Senses

The, more or less, unconventional way this thesis has been conducted (*i.e., two experimental areas: Greek and English*), offers us the "right" to spend a few lines about senses – it won't be long!
So, what does a sense? What is it? Where did it come from? How does it work? What's its contribution?... The answer is simple: A sense is, intuitively, a context region, a word area which has some unique characteristics. In our case, our senses composed by words of high frequency – with other words, words that are repeated inside this context group at least often.
Consider a mesh of repeated and repeated word garbages... Now, forget it! Senses will do the job! They will collect the noise (if any) inside one or more context groups, leaving other contexts clear and noise-free in order to be used from context metrics; and of course we apply an algorithm which distinguishes and finds the noise-free senses. However, the senses-idea doesn't necessary lean on garbages! Anything but that, the claim is generally that we are trying to identify a number of regions, word groups in order to apply our metrics between desired combinations (of clusters –senses–) or, maybe, between, e.g., two prevailed areas, which achieve the highest similarity score to a word[5].

[5]Look for more in chapter 3

1.7 Thesis outline

Next chapters gradually reveal the "body" of the thesis, which, concisely, is contained of: 1) *the metrics*; the metrics we use in order to calculate the similarities between contexts, 2) *the algorithms description*; for both Greek datasets and English datasets, 3) *the datasets and questionnaires* we use (as far as the Greek datasets is concerned) in order to "pick up" the human similarities, 4) *the Experiments* which are conducted for both Greek and English pairs, but with a different experimental procedure, and finally, 5) *the experimental conclusions* of our work.

2 Semantic Similarity Metrics

2.1 Introduction

In this chapter we present you the metrics we use in order to calculate the semantic similarity between two words. First and foremost, by saying semantic similarity we mean the semantic relatedness, which mean: how much does term A have to do with term B?
There are two "metric groups": the *Page-count-based Similarity Metrics* (as in [6]) and the *Fully text-based Similarity Metrics*. Both metrics calculate, apparently, as we said, a semantic similarity grade. Their differences are important.
Firstly, note that both types are actually *unsupervised* and *web-based*, which means that they are independent of any king context databases or word-lexicons, like, e.g., WordNet; their only source is the web, the sites in which the keywords of a pair appear.
As far as the page-count-based similarity metrics are concerned, consider, as their name denotes, that they take advantage only the number of pages which a web search engine[6] returns, while the fully text-based similarity metrics make use only of the contexts of downloaded documents.
Their applications can be quite many, e.g.[7]: Consider Amazon; consider when you pick your new favorite book, background semantic similarity metrics running; imagine codes to scan your beloved book in order to find keywords (probably with high frequency); imagine the same codes to scan other books (probably of the same category) in order to find similar contexts... i.e. resembling words... i.e. similar books. And then, at next page... you have not only the common: "customers who bought this item also bought bla-bla-bla", but "unusual" semantic suggestions as well! – Similar applications we can have on web search engines, YouTube-like sites, e-shops, etc.
At next paragraphs we examine thoroughly both types.

2.2 Page-count-based similarity metrics

Page-count similarity metrics take advantage of, as we mentioned above, the number of pages that a web search engine returns in order of do a quick similarity approximation. The basic idea under this approach is that the word

[6] Yahoo Search API in our thesis
[7] A very interesting example which we referred in chapter 1

co-occurrence is likely to indicate some kind of semantic relationship between words. However, the number of documents in which a certain word pair co-occurs, does not express a direct semantic similarity. Additionally, it is reasonable also to take into account the number of documents that include each pair component individually for normalization purposes. In other words, for a word pair, we need to know the information that the two words share, normalized by the degree of their independence.
The page-count metrics we use are three: Jaccard, Dice and Mutual Information (MI). In order to understand the metrics, it is necessary to display the definitions below [7]:

{D}: a set containing the whole document collection that are indexed and accessible by a web search engine
$\mid D \mid$: the number of documents in collection {D}
w_i: a word or term
$\{D \mid w_i\}$: a subset of {D}, documents indexed by w_i
$\{D \mid w_i, w_j\}$: a subset of {D}, documents indexed by w_i and w_j
$f(D \mid w_i)$: the fraction of documents in {D} indexed by w_i
$f(D \mid w_i, w_j)$: the fraction of documents in {D} indexed by w_i and w_j

2.2.1 Jaccard metric

The Jaccard coefficient is a measurement calculating the similarity (or diversity) between sets. Particularly, we use a variation of the Jaccard coefficient whose formula is defined as:

$$Jaccard(w_i, w_j) = \frac{f(D \mid w_i, w_j)}{f(D \mid w_i) + f(D \mid w_j) - f(D \mid w_i, w_j)} \tag{1}$$

In probabilistic terms, equation 1 calculates the maximum likelihood estimate of the ratio of the probability of finding a document where words w_i and w_j occurs. When w_i and w_j are the same word, then the Jaccard coefficient is equal to 1 (absolute semantic similarity), and when the words never co-occur in a document, then the Jaccard coefficient is 0.

2.2.2 Dice metric

Dice coefficient which is related to the Jaccard coefficient. Its formula:

$$Dice(w_i, w_j) = \frac{2f(D \mid w_i, w_j)}{f(D \mid w_i) + f(D \mid w_j)} \tag{2}$$

Similarly, when w_i and w_j are the same word, Dice coefficient is equal to 1, and when the two words never co-occur, coefficient is equal to 0.

2.2.3 MI metric

Considering the occurrences of words w_i and w_j as random variables X and Y, respectively, the Mutual Information among X and Y measures the mutual dependence between the appearance of words w_i and w_j. The maximum likelihood estimate of MI is:

$$MI(X, Y) = log \frac{\frac{f(D|w_i, w_j)}{|D|}}{\frac{f(D|w_i)}{|D|} \frac{f(D|w_j)}{|D|}} \tag{3}$$

Mutual information measures the information that variables X and Y share. It quantifies how the knowledge of one variable reduces the uncertainty about the other. For instance, if X and Y are independent, then knowing X does not give any information about Y, and the mutual is 0. For X=Y, the knowledge of X gives the value of Y without uncertainty and the mutual information is 1. Note that the fractions of documents are normalized by the number of documents indexed by the search engine, $\mid D \mid$, giving a maximum likelihood estimate of the probability of finding a document in the web that contains this word.

2.2.4 NGD metric

Google distance is a measure of semantic interrelatedness derived from the number of hits returned by the Google search engine (consider Yahoo Search Engine for us) for a given set of keywords. Keywords with the same or similar meanings in a natural language sense tend to be “close” in units of Google distance, while

words with dissimilar meanings tend to be farther apart. Specifically, the normalized Google distance between two search terms x and y is:

$$NGD(x,y) = log\frac{max\{logf(x), logf(y)\} - logf(x,y)}{logN - min\{logf(x), logf(y)\}} \quad (4)$$

The N means the total number of web pages searched by Google (Yahoo in this thesis); f(x) and f(y) are the number of hits for search terms x and y, respectively; and f(x, y) is the number of web pages on which both x and y occur.
If the two search terms x and y never occur together on the same web page, but do occur separately, the normalized Google distance between them is infinite. If both terms always occur together, their NGD is zero. Consequently, the wanted similarity is given by:

$$sim(x,y) = e^{-2 \cdot NGD(x,y)} \quad (5)$$

2.3 Fully text-based similarity metrics

One semantic similarity metric, which is a variation of the cosine similarity metric, is used in order to measure the semantic distance between words and in order to automatically generate semantic classes. This metric, CS^W_{WS}, computes wide-context similarity between words using a bag-of-words model, while other metrics (such as CS^N [3]) compute narrow-context similarity using a bigram language model. Such metrics rely on the idea that similarity of context implies similarity of meaning [1]. We assume that words which appear in similar lexical environment (e.g., inside a web site context), have a close semantic relation [1,8,9].
In bag-of-words models [10,11], as we have already mentioned, we apply a WS in order to grab the neighboring words (–for a specific keyword– which are located either on the left side or on the right side, i.e. one WS is applied in the left side while another one in the right side). Now, we assume: $[v_{WS,L}...v_{2,L}v_{1,L}]$ and

$[v_{WS,R}...v_{2,R}v_{1,R}]$ as left and right contexts (of a word *w*) correspondingly, i.e. $v_{i,L}$ and $v_{i,R}$ mean the i^{th} the left and the right word correspondingly of a word *w*. The feature vector for every word *w* is defined as $T_{w,WS} = (t_{w,1}t_{w,2}...t_{w,N})$ where $t_{w,i}$ is a non-negative integer. The feature vector size is equal to the vocabulary size N, i.e, there is a feature vector for every word of the vocabulary. Now, the question is that: What does this vector represent? The i^{th} feature value $t_{w,i}$ shows the occurrences of vocabulary word v_i within the left or right context of a word *w*, and is set according to a *Binary* a *Frequency* or a *Logarithmic Frequency Scheme*.

2.3.1 Binary cosine metric

The *binary cosine metric* assigns 1 if the v_i appears (one or more times) within the left and right context of a word *w*, and assigns 0 if the v_i does not exist inside both contexts (left and right). The coefficient is defined as:

$$CS_{WS}^{W} = \frac{\sum_{i=1}^{N} t_{w_1,i}^{bin} t_{w_2,i}^{bin}}{\sqrt{\sum_{i=1}^{N}(t_{w_1,i}^{bin})^2}\sqrt{\sum_{i=1}^{N}(t_{w_2,i}^{bin})^2}} \tag{6}$$

2.3.2 Frequency cosine metric

The *frequency cosine metric* assigns the number of occurrences of v_i within the left and right context of a word *w*, and assigns 0 if the v_i does not exist inside both contexts (left and right). The coefficient is defined as:

$$CS_{WS}^{W} = \frac{\sum_{i=1}^{N} t_{w_1,i}^{freq} t_{w_2,i}^{freq}}{\sqrt{\sum_{i=1}^{N}(t_{w_1,i}^{freq})^2}\sqrt{\sum_{i=1}^{N}(t_{w_2,i}^{freq})^2}} \tag{7}$$

2.3.3 Logarithmic frequency cosine metric

The *logarithmic frequency cosine metric* is similar to *frequency cosine metric*. The difference is that we calculate the logarithm of frequency of word occurrences of word, e.g., v_i. Note that log(0) does not exist! For that reason, in such a case, we assign a value, e.g., 10^{-6}. The coefficient is defined as:

$$CS_{WS}^{W} = \frac{\sum_{i=1}^{N} t_{w_1,i}^{logfreq} t_{w_2,i}^{logfreq}}{\sqrt{\sum_{i=1}^{N} (t_{w_1,i}^{logfreq})^2} \sqrt{\sum_{i=1}^{N} (t_{w_2,i}^{logfreq})^2}} \tag{8}$$

3 Semantic Similarity Computation Algorithm

3.1 Introduction

In this chapter we present the algorithmic procedure which we use in Greek and English datasets in order to calculate the similarities and, moreover, the correlations. The algorithms, apparently, are not the same, as in one case (English datasets) we work with senses, while in the other case we don't. Additionally, in every step we may also spend a few more lines if we count as necessary to clear up any details.

3.2 Context-based similarity algorithm

The Context-based Similarity Algorithm is, we could say, our baseline algorithm; actually it is the same algorithm with that in paper [3]. The only slight difference is that of our filter[8].– Our motivation, our goal, as we mentioned in chapter 1, is to invent an algorithmic procedure which will calculate the semantic similarity between words automatically, i.e. unsupervised. There are algorithms, like WordNet-based algorithms, i.e. supervised algorithms, that they are human dependent procedures. Thus, this algorithm offers us the potential ability not to maintain, support and manage huge and complex e-lexicons...–Finally, here it is, the algorithm:

STEP 1: Before running experiments (i.e. metrics implementation on contexts, and estimation of semantic-similarity correlation between humans and computers) it is needed, first, a number of contexts to be downloaded. The procedure is simple: For every pair of a dataset (one of the two) we grab, e.g., 100 URLs (using Yahoo Searsh API), and then, we get their HTMLs, i.e. 100 contexts for each pair.– Epigrammatically, these are the sub-steps;

- FOR EVERY pair { download the URLs }
- FOR EVERY pair { download the HTMLs }

STEP 2: Since we have downloaded the HTMLs, we continue to the next step; filtering takes place. We apply two filters: One *HTML filter*, which removes the

[8]Due to of Greek language.

HTML tags, and one more, the so called *TURBO filter* (our conception...), which removes: numbers, latin letters, symbols (such as @?$+), capital letters, capital words and word repetition. – We remove the capital letters, i.e., e.g. "Nick Nick" becomes "Nick". The reason we do this is because in PERL[9] doesn't support *Greek Case Insensitive....* Thus, the Greek word, e.g., "Σκύλος" becomes "σκύλος". Moreover, we also remove the capital words. In Greek language, in contrast with English, there are tones (obviously, in a different way than that of English) which are not kept when the word is written with capital letters. This difference raises the next problem: In case we meet a capital word inside the context, we can't make it non-capital, simply because we don't know where the tone has been placed. E.g., the word "ΑΥΤΟΚΙΝΗΤΟ" would become "αυτοκίνητο" (i.e. without any tone) instead of the right "αυτοκίνητο" (with a tone)! The difference between the two words is that we can grab the word "αυτοκίνητο", while we can't the word "αυτοκινητο"[10]. For that reason we remove capital words. Finally, since we have organized the filtered data into the right "experimental folders", we run the metrics on them. The script grabs actually the occurrences, for both words of a pair, inside the contexts (by applying a WS), and, then, calculates the similarity.– Epigrammatically, these are the sub-steps;

- ON EVERY pair context { apply the HTML filter }
- ON EVERY html-filtered context { apply the TURBO filter }

STEP 3: Since we have organized the filtered data into the right "experimental folders", we apply the metrics on them. A "metric-script" grabs actually the occurrences for both words of a pair inside the contexts (by applying a WS), and then calculates the similarity.– Epigrammatically, these are the sub-steps;

- WS = input
  FOR EVERY word of a pair
  { grab the occurrences }
  calculate the pair similarity

STEP 4: Finally, the last step concerns the correlation calculation. We have inserted both computer and human similarity grades inside a correlation-calculation

[9]The language we, mainly, use in this thesis.

[10]...to grab a word in order to calculate the semantic similarity. We want to say i.e. that a word without a tone doesn't help the metrics...

file and, now, all we have to do is to push the run-button...– Epigrammatically:

- Correlation calculation

3.3 Word senses based algorithm

As far as the Word Senses based Algorithm is concerned, we could say that is a branch of the Context-based Similarity Algorithm. Our motivation, our goal is to invent an even more effective algorithmic procedure, than the previous which was analyzed, in order to calculate, of course, the semantic similarity between two words. The idea is quite simple. As we mentioned earlier in chapter 1, what we do is to cluster a context into smaller pieces, e.g. 3, with common characteristics. Since we have done it for the contexts of both words, we can try of "cluster combinations". Specifically, in this algorithm, we approach the "strongest-sense idea".– Here it is, the algorithm:

STEP 1: Initially, for every pair of the dataset, as also for every single (unique) word of a dataset, we grab e.g. 100 URLs –using Yahoo Search API–, and then, we download their HTMLs, i.e. 100 contexts for each pair and for each unique word.– Epigrammatically, these are the sub-steps;

- FOR EVERY pair AND FOR EVERY unique word{ download the URLs }
- FOR EVERY pair AND FOR EVERY unique-word{ download the HTMLs }

STEP 2: Since we have downloaded the HTMLs, we continue to the next step; filtering takes place. We apply two filters: One *HTML filter*, which removes the HTML tags, and one more, the so called *TURBO filter* (our conception...), which removes: numbers, symbols (such as @?$+), capital letters (i.e. we alter them to non-capitals, we do not delete them), and word repetition.– As far as the word repetition is concerned, we present the following example: the "Nick Nick" becomes "Nick". Note that despite there is not any problem with *English Case Insensitive...*, we alter capital letters to small ones for the sake of grace!

STEP 3: Now, the next step is to grab the occurrences applying a WS; and this is done for both "one-word" and "two-word contexts", i.e. for pair-of-words contexts as also for single-word contexts.– Epigrammatically, these are the sub-steps;

- ON EVERY context { apply the HTML filter }
- ON EVERY html-filtered context { apply the TURBO filter }

STEP 4: Since we have isolated the occurrences *(particularly, at this step, we are interested only to "one-word contexts")*, we move on to the encoding; we encode our occurrences in order to split them, to cluster them. The encoding method is this: For every line (i.e. for every occurrence –consider that we have lines of occurrences–) we count the frequency of every unique word; then, we write a new line (in another file) which composed by a sequence of numbers, where every number corresponds to a unique word, which single word belongs to the whole context! I.e. each line contains all frequencies of all the words of the context, but the numbers which are not equal to zero, correspond, apparently, to the words of the specific line!– Epigrammatically, for unique-word contexts...:

- grab the occurrences
- encode the occurrences

STEP 5: Moreover, since we have done with encoding, next step is clustering. How does this take shape? It's simple! Consider the grand matrix we have just made. Since we have appended it into the right program[11], we are ready to run the sense-script. But, how many clusters do we permit? and, particularly, which is our "cluster-criterion"?– Good questions! As far as "how many clusters we permit", we make a convention: maximum clusters, 3. As for "cluster-criterion", we use the entropy as "impurity pointer", i.e. how much "word sameness" a cluster includes.– Epigrammatically:

- cluster unique-word encoded contexts

STEP 6: Since we have done with the clustering above, we decode data we have clustered in order to take back the "real" occurrences.– Epigrammatically:

- decode clustered contexts

[11] Consider, e.g., MATLAB.

STEP 7: Now, at this step we cluster the "two-word contexts" (or, with other words, pair-of-words contexts). However, this clustering differs from the previous one.– This clustering is implemented as follows: For every word of a pair and for every occurrence of this word, we apply a context-based similarity metric between this occurrence and a whole sense (apparently, a sense of the correspondent single-unique word), and this is done for all the senses (of the single-unique word). The sense which "hits" a high similarity score "tells" to the "pair occurrence" to be "grouped" to a new correspondent sense...– Epigrammatically, for pair contexts:

- FOR EVERY occurrence of a pair context
  { apply a context-based metric between the occurrence and the senses of the corresponding unique word contexts }
  Find the high score similarity AND classify occurrence to the right sense

STEP 8: Since we have done with second clustering, we calculate the similarities. The procedure is the following: First and foremost we have a pair of words, where every word has a number of senses. What we do is to apply a context-based similarity metric between all the combinations of the senses (of each word). The senses which achieve a high similarity score are considered as "strong" and their similarity is being saved. The procedure is repeated for all pairs.– A senses figure example is presented below:

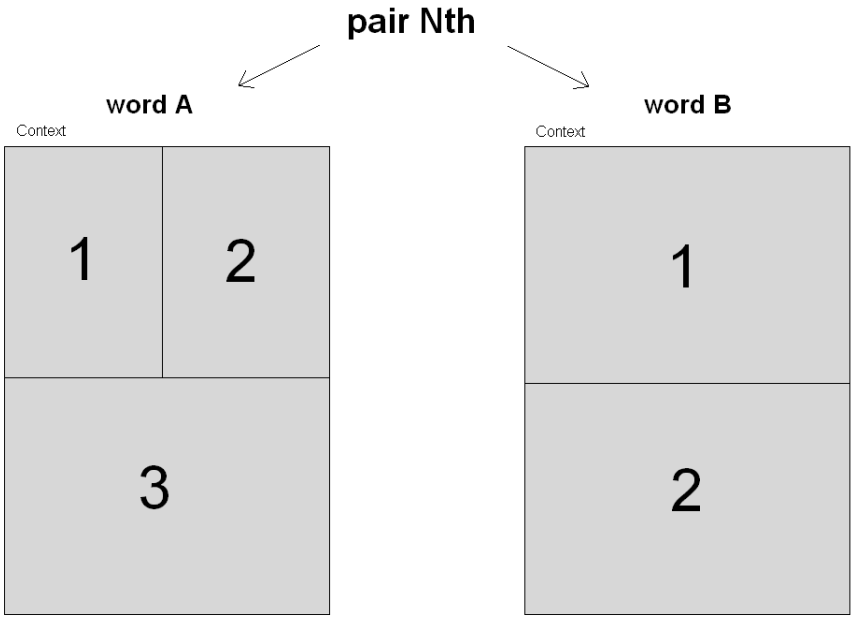

Fig.17 A senses example

Epigrammatically, the sub-steps are:

- FOR EVERY pair
  { apply a context-based metric between wordA senses and wordB senses AND save the highest similarity score }

STEP 9: Finally, the last step concerns the correlation calculation. We have inserted both computer and Charles-Miller similarity grades[12] inside a correlation-calculation file and, now, all we have to do is to push the run-button...– Epigrammatically:

- correlation calculation

[12]The grades which put the Charles-Miller subjects...

4 Datasets and Questionnaire

4.1 Introduction

This chapter concerns the Greek datasets of our thesis, and particularly the dataset and questionnaire construction.– In order to calculate the "semantic correlation" between humans and computers, we need first to have a dataset to work with. Precisely, we make two datasets, of course in Greek. Both datasets construction is based on the Charles-Miller dataset [2].– Since we have done with our datasets, we create two questionnaires in which datasets will be appended; i.e., more clearly: the first questionnaire takes the first dataset, while the second questionnaire takes the other one.

4.2 Charles-Miller dataset

In paper [2] Charles and Miller use a subset of 30 noun pairs from the original list of 65 studied in paper [1]. Our datasets are based on this dataset.– Charles and Miller selected their pairs as follows: They selected 10 pairs from the high semantic-similarity level, 10 from the intermediate level, and 10 from the low level. Then, the 30 test pairs were printed on two separate sheets of paper in order to be distributed to two different groups.– The dataset is being presented below:

Ch. & M. dataset		
1.	car	automobile
2.	gem	jewel
3.	journey	voyage
4.	boy	lad
5.	coast	shore
6.	asylum	madhouse
7.	magician	wizard
8.	midday	noon
9.	furnace	stove
10.	food	fruit
11.	bird	cock
12.	bird	crane
13.	tool	implement
14.	brother	monk
15.	lad	brother
16.	crane	implement
17.	journey	car
18.	monk	oracle
19.	cemetery	woodland
20.	food	rooster
21.	coast	hill
22.	forest	graveyard
23.	shore	woodland
24.	monk	slave
25.	coast	forest
26.	lad	wizard
27.	cord	smile
28.	glass	magician
29.	rooster	voyage
30.	noon	string

Tab.1 Charles & Miller dataset

The dataset consists of 4 synonyms, 3 hypernyms, 6 hyponyms, 3 coordinate terms and 12 more uncategorized pairs; however, there isn't any "opposite pairs", i.e., e.g. "dependent - independent".

Note that Charles and Miller use 30 pairs, while in [3] they use 28. The reason is that since they have calculated the correlation results (in [3]) they compare them to the correspondent supervised methods, and they can do that only for the 28 pairs!

4.3 Thesis datasets

Before we figure out the "semantic correlation" between human beings and machines, we first need, of course, to create a dataset, and, specifically, two datasets. Each dataset consists of 30 Greek pairs, which are all, basically, nouns. But, why to choose two datasets?
Well, the first dataset was composed, mostly, as a precise translation of Charles and Miller (Ch. & M.) dataset [2], while the other set is considered to be a conception of ours! Concisely, we refer that also the dataset of Ch. & M. was based on that of paper [1] of Rubenstein and Goodenough (R. &. G.) – it is a quite large datasets of 65 pairs. This is the "dataset base" of Ch. & M., and, furthermore, our base. But let's take a look of our datasets.
As far as the first mentioned set is concerned, it was created by our curiosity to examine how the English-Greek translation works; i.e., if we can achieve a similar correlation with that of [3]. Note that most of the synonyms, hyponyms, meronyms, etc. maintained after translation, as also the *multi-meanings* a word may have.
As far as the second set is concerned, it is based on Ch. & M. set; i.e. we have "provided" it with hyponyms, hypernyms, synonyms, coordinate terms, etc.

Below, we present you our sets:

OUR 2 DATASETS					
1.	αμάξι	αυτοκίνητο	1.	μόλυνση	ρύπανση
2.	πετεινός	κόκκορας	2.	κόμμα	παύση
3.	βόλτα	περίπατος	3.	τιμή	κόστος
4.	αγόρι	νεαρός	4.	νοημοσύνη	ευφυΐα
5.	ακτή	παραλία	5.	δεσμός	σχέση
6.	ψυχιατρείο	τρελοκομείο	6.	δρόμος	μονοπάτι
7.	δούλος	σκλάβος	7.	χρήμα	κεφάλαιο
8.	προφήτης	μάντης	8.	σκέψη	φιλοσοφία
9.	μάντης	μάγος	9.	σχέση	σύντροφος
10.	τροφή	φρούτο	10.	διαμέρισμα	σπίτι
11.	πουλί	χελιδόνι	11.	παντελόνι	ρούχο
12.	πουλί	γερανός	12.	βιβλίο	σελίδα
13.	όργανο	εργαλείο	13.	σελίδα	χαρτί
14.	αδελφός	μοναχός	14.	κεφάλαιο	μέτοχος
15.	νεαρός	αδελφός	15.	βιβλίο	κεφάλαιο
16.	γερανός	εργαλείο	16.	πρώτος	αριθμός
17.	βόλτα	αμάξι	17.	συγκρότημα	τραγουδιστής
18.	μοναχός	προφήτης	18.	σύμπαν	διάστημα
19.	κοιμητήριο	δάσος	19.	πρώτος	καλύτερος
20.	τροφή	κόκκορας	20.	συγκρότημα	διαμέρισμα
21.	ακτή	λόφος	21.	κόμμα	βουλή
22.	δάσος	νεκροταφείο	22.	κόμμα	τελεία
23.	παραλία	δάσος	23.	κόστος	κεφάλαιο
24.	μοναχός	δούλος	24.	υδρογόνο	δεσμός
25.	ακτή	δάσος	25.	μόλυνση	χαρτί
26.	νεαρός	μάγος	26.	δρόμος	διαμέρισμα
27.	αγόρι	πετεινός	27.	απορία	σχολείο
28.	ψυχιατρείο	φρούτο	28.	μόλυνση	σχολείο
29.	κόκκορας	περίπατος	29.	σύμπαν	βουλή
30.	αυτοκίνητο	μάγος	30.	αυτοκίνητο	ρούχο

Tab.2 Our two datasets

The left-one dataset presents the Ch. & M. translation, while the right-one presents the "new-one" (our conception...).

Moreover, additionally to the datasets above (which concern the Greek datasets), we have one more dataset in English, for English datasets. This dataset is shown below:

Ch. & M. dataset		
1.	car	automobile
2.	gem	jewel
3.	journey	voyage
4.	boy	lad
5.	coast	shore
6.	asylum	madhouse
7.	magician	wizard
8.	midday	noon
9.	furnace	stove
10.	food	fruit
11.	bird	cock
12.	bird	crane
13.	tool	implement
14.	brother	monk
15.	lad	brother
16.	crane	implement
17.	journey	car
18.	monk	oracle
19.	food	rooster
20.	coast	hill
21.	forest	graveyard
22.	monk	slave
23.	coast	forest
24.	lad	wizard
25.	cord	smile
26.	glass	magician
27.	rooster	voyage
28.	noon	string

Tab.3 Iosif & Potamianos dataset

This dataset consists of 28 parts and it is the same dataset which use in [3]. As we said before in [3] they use 28 pairs out of 30 (which the original has [2]). We repeat that the reason they do that is because when they calculate the correlation results, they compare them to correspondent supervised ones... and they can do that only for the 28 pairs!

4.4 Questionnaires

As long as we have constructed the two sets, we get work in order to create a questionnaire; particularly, two questionnaires; one for each dataset.
First and foremost, as we have already said, in each questionnaire will be appended one dataset (one of the two).
The questionnaire-construction purpose is, apparently, the semantic-similarity rating of the pairs. Thus, we put a *semantic-similarity grade-scale*, which is between 0 and 4; namely: $0-1-2-3-4$. The reason we have chosen a 5-grade semantic similarity scale is due to the "distinguishing"' limitation of the human brain.
Human brain's "distinguishiality" is, definitely, not infinity... Human brain, instinctively, can distinguish-separate, e.g., three objects putted randomly in a table. A concentrated human brain can distinguishes 5 objects! A super brain may achieve a seven, or maybe a ten! Certainly, a casual man, is a 5-man! But, lets get back in questionnaires.
We have put each datasets in a questionnaire, we have chosen a semantic grade scale, and, we collect some extra information about the person who rates; such an information are: *age, gender, education and occupation*. Now, what's the final step? The questionnaire distribution.
All questionnaires have been distributed at 30 subjects (mainly students), and each subject will be given two questionnaires (one for each set).

4.5 Subject information

Since we have distributed and collected the questionnaires, we process the data. Initially, we present you the basic information about the subjects who filled the questionnaires.

Our first figure depicts the subjects "age histogram":

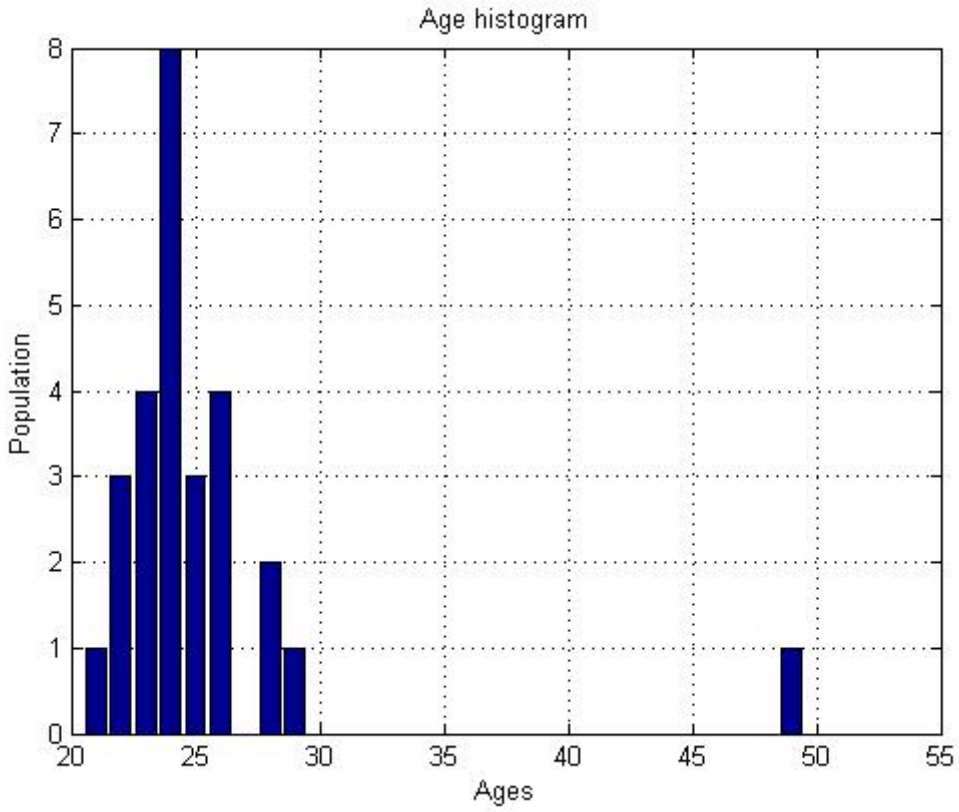

Fig.1 Age histogram

"Accidentally", this figure looks like a Gaussian... Most of the ages located between 21 and 29 years, and prevailed appears to be the 23 years. The mean age equals to 23.47, and without the 49-year peak, equals to 22.59.
The same subject target-group chose on [1], where all subjects were students[13]. A difference between our thesis (in this part) and the [1] is that the students were not paid!

Next figure depicts a "gender histogram":

[13] In our case most of them are students.

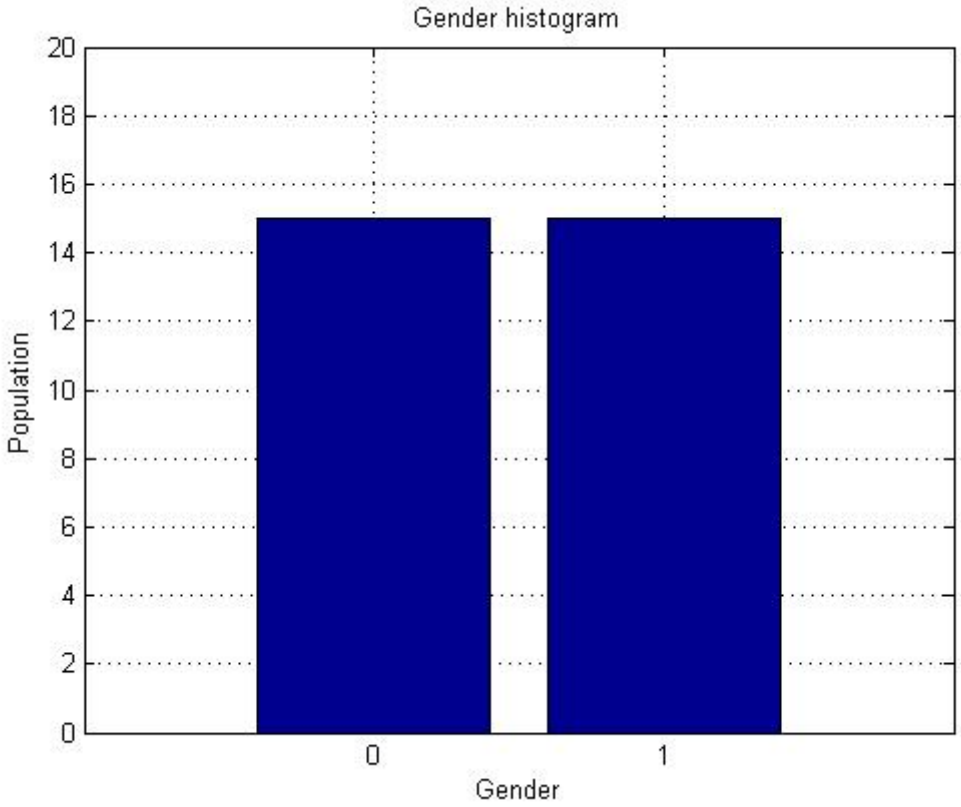

Fig.2 Gender histogram

Who's a man and who's woman:
0: man
1: woman

Actually… this diagram shows that 50% are men and 50% are women. This happend also accidentally – definitely, it was't in our plans.What to say!

Next figure is an "education histogram":

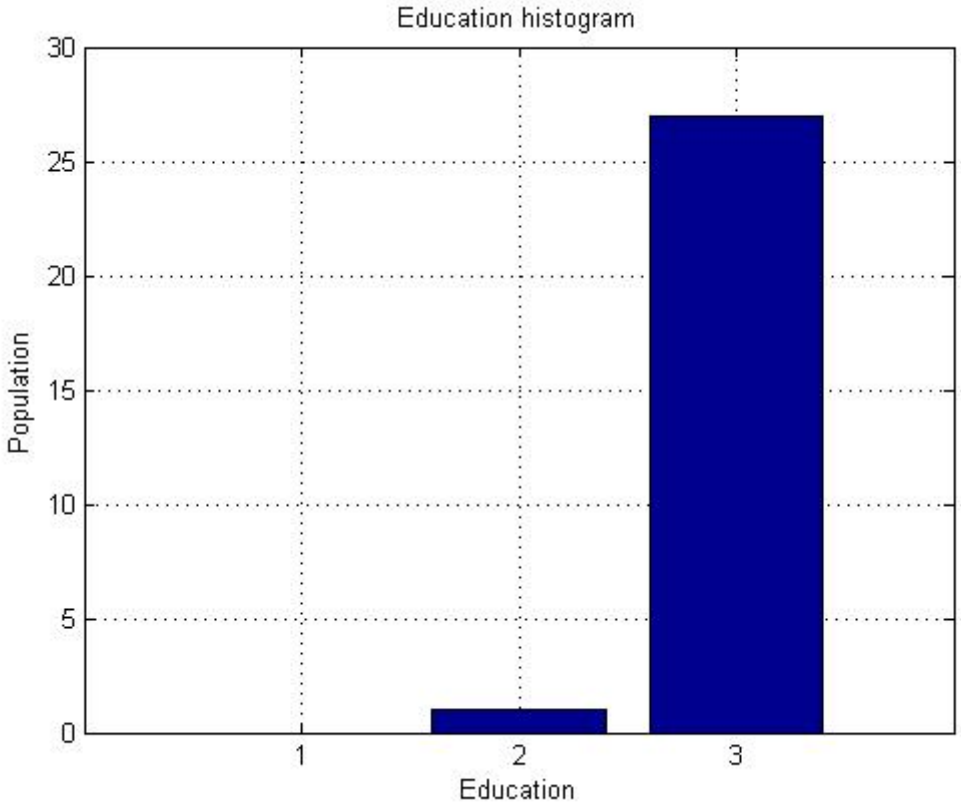

Fig.3 Education histogram

What's their education:
1: primary school education
2: high school education
3: university education

All subjects appear to have university education. However, two of them seem to have high school education. We believe that this is not true (!) as most of the students have not completed their studies, some of them may still feel young!

Next figure is an "occupation histogram":

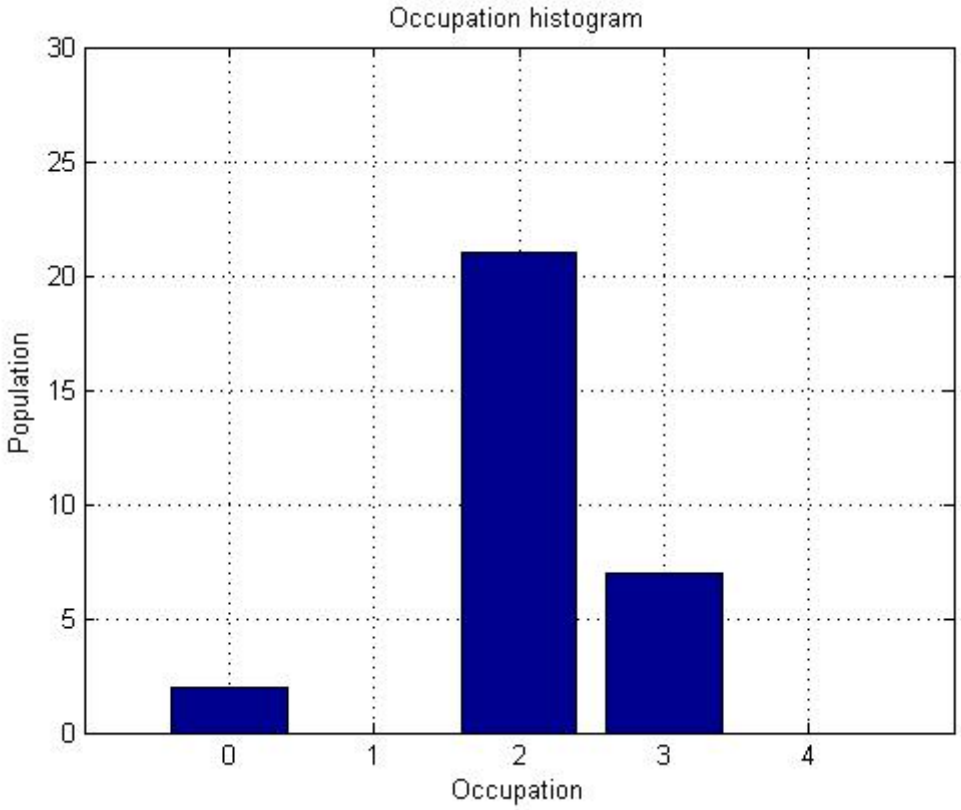

Fig.4 Occupation histogram

What's their occupation:

1: pupil

2: student

3: worker

4: retired

0: other

The figure shows that "student" is the prevailed occupation, while second appear to be "worker".

5 Experiments

5.1 Introduction

In this chapter we will present you the experimental part of the thesis. The experiments are divided into two parts: one concerns Greek datasets), and one the English dataset. In both parts, as we have mentioned many times before, we will, mainly, apply the context-based metrics (as also, of course, the page-counts ones[14]). The difference between the two parts is the experimental approach; we, concisely, repeat that in English datasets we use senses[15], while in Greek datasets we use only two contexts to compare... or, with other words, two senses! Finally, in each case, in each part we try more than one experimental approaches.

5.2 Greek datasets

Since we have completed the pre-experimental procedure, we are now ready to launch and present you our experiments. We have collected the questionnaire data, we have stored them in computer, we have processed subjects' "life information" (age, gender, education and occupation), as also their *similarity grades*, and, finally, we are ready! Note that *the following experiments run on 100 URLs and 300 URLs*.

5.2.1 100 URLs

Our first figure, below, presents the alteration of corellation by Window Size of "our dataset"; the metrics that applied are three and they are context-based. Also, the contexts have not been extra-filtered with TURBO-filter, they are only HTML-filtered. Here it is, our first figure:

[14] Especially, in Greek datasets.

[15] More than one.

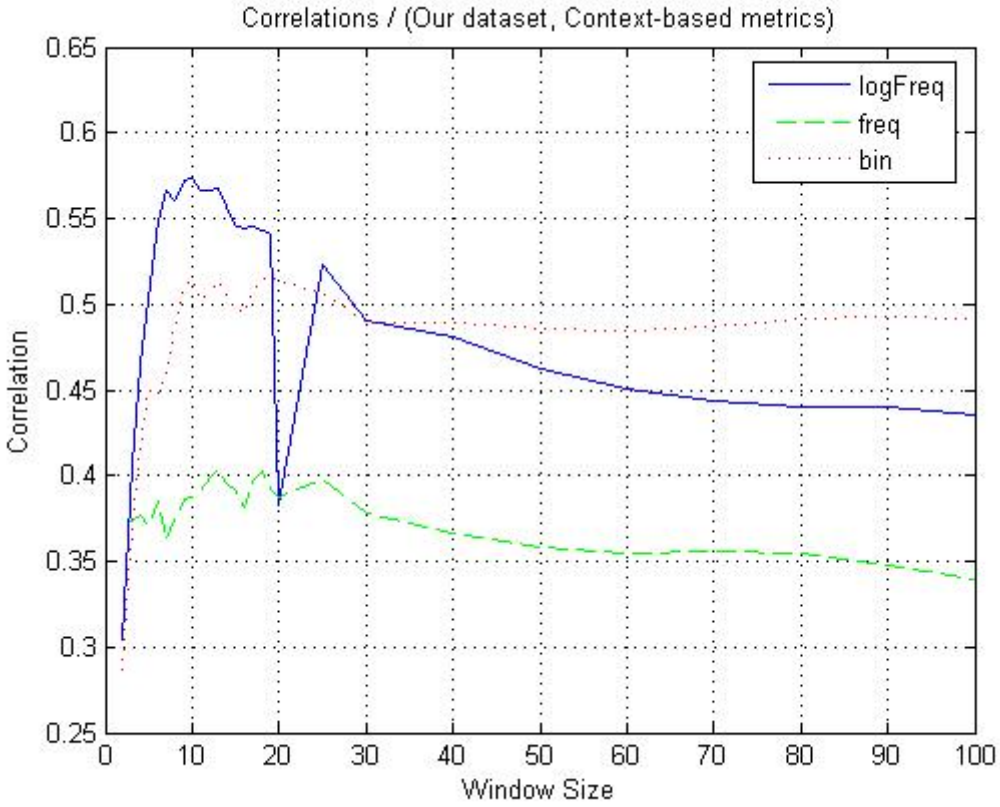

Fig.5 Correlations / (100 URLs, non-translated dataset, context-based metrics, not TURBO-filtered contexts)

We notice that best metric seems to be logFreq, which achieves a high correlation score on WS 10, and equals to 0.57. However, a peak on WS 20 doesn't make us very happy. Continuing, as WS increases, logFreq seems to decline. Note that for small window sizes logFreq (as also the other metrics) achieve a small correlation score, which is a totally different behavior comparing to fig.1 in [3][16]. Moreover, bin metric seem to have a more or less stable behavior for $WS > 10$, while freq appears way down from the other ones. But let's see another similar figure; the difference, now, is that the contexts have been TURBO-filtered:

[16]Note that the correspondent figure of fig.1 in [3] is presented next

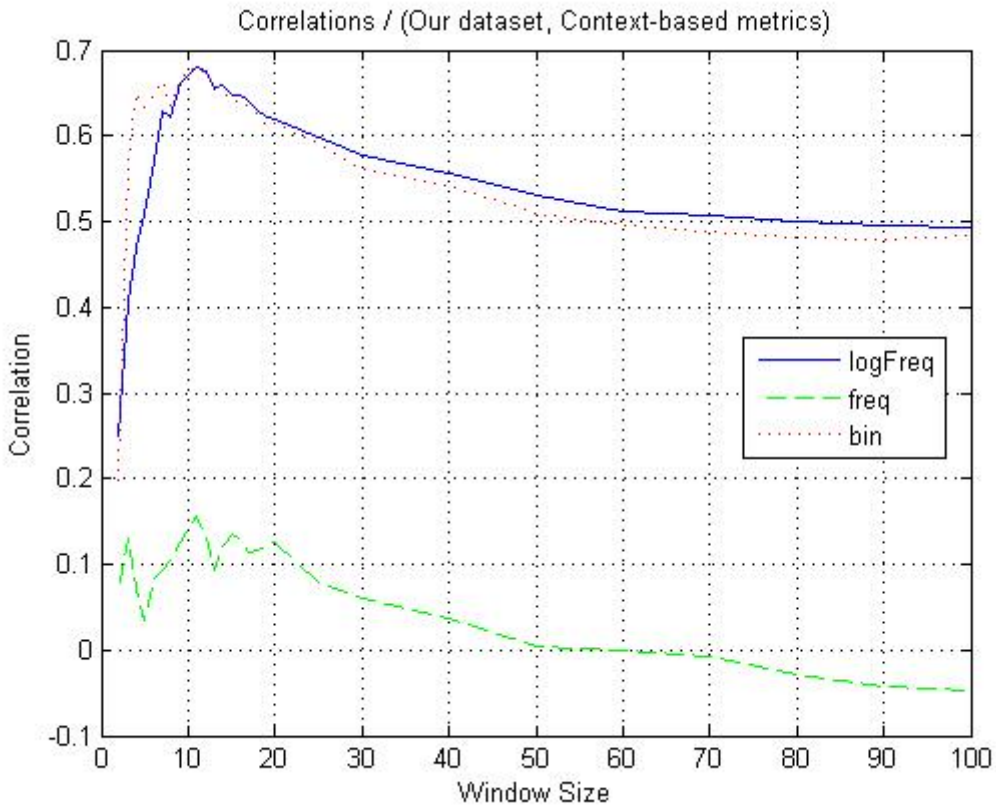

Fig.6 Correlations / (100 URLs, non-translated dataset, context-based metrics, TURBO-filtered contexts)

Now, things seem to be much better. The prevailed metrics, this time, seem to be actually both logFreq and binFreq, which achieve together (!) a high correlation of 0.68 on WS 11, on the same WS, as before on fig.5. Also, the peak we had noticed before has gone and the functions (bin and logFreq) appears quite peaceful! Nevertheless, again, we notice that for $WS < 10$ we still have small (actually, even smaller) correlation scores. Moreover, the freq metric appears to have dangerously been declined "touching" the zero correlation... Lastly, all metrics appeared, again in this diagram, to decline as window size increases – tottaly different behavior, as we said before, from the truly correspondent, this time, fig.1 in [3].

Now, it's time to see how the page-count metrics work for both these cases.

At next figures, each metric belongs to a number; so, in order to understand the figures...:

1: NGD metric

2: MI metric

3: Dice metric

4: Jaccard metric

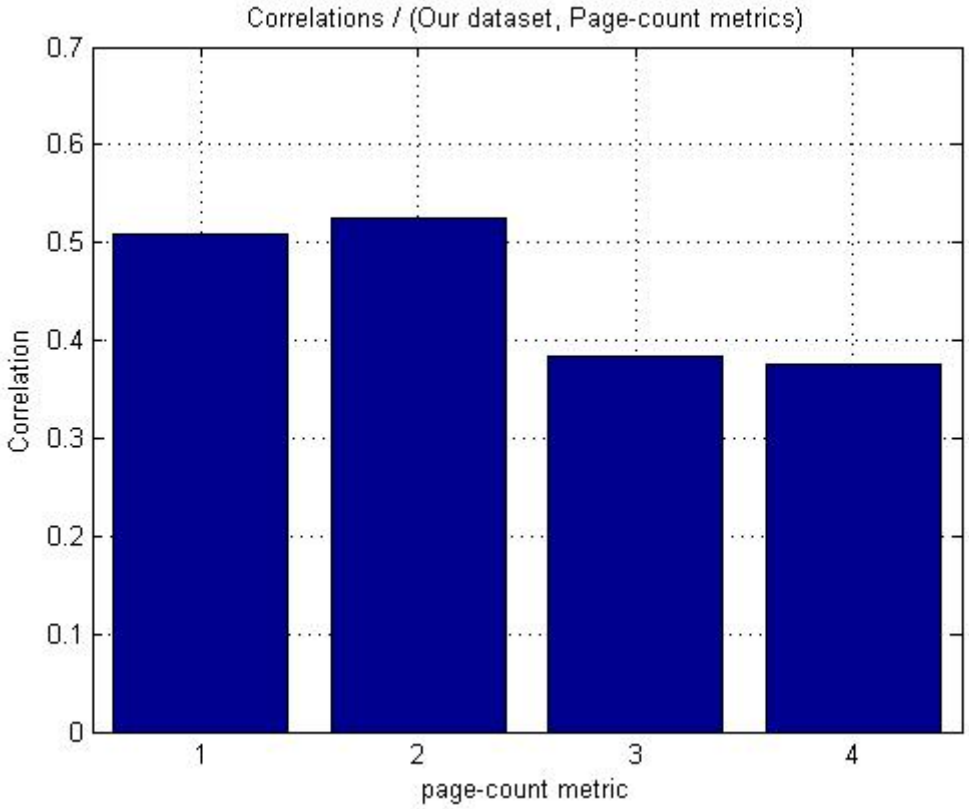

Fig.7 Correlations / (100 URLs, non-translated dataset, page-count metrics, not TURBO-filtered contexts)

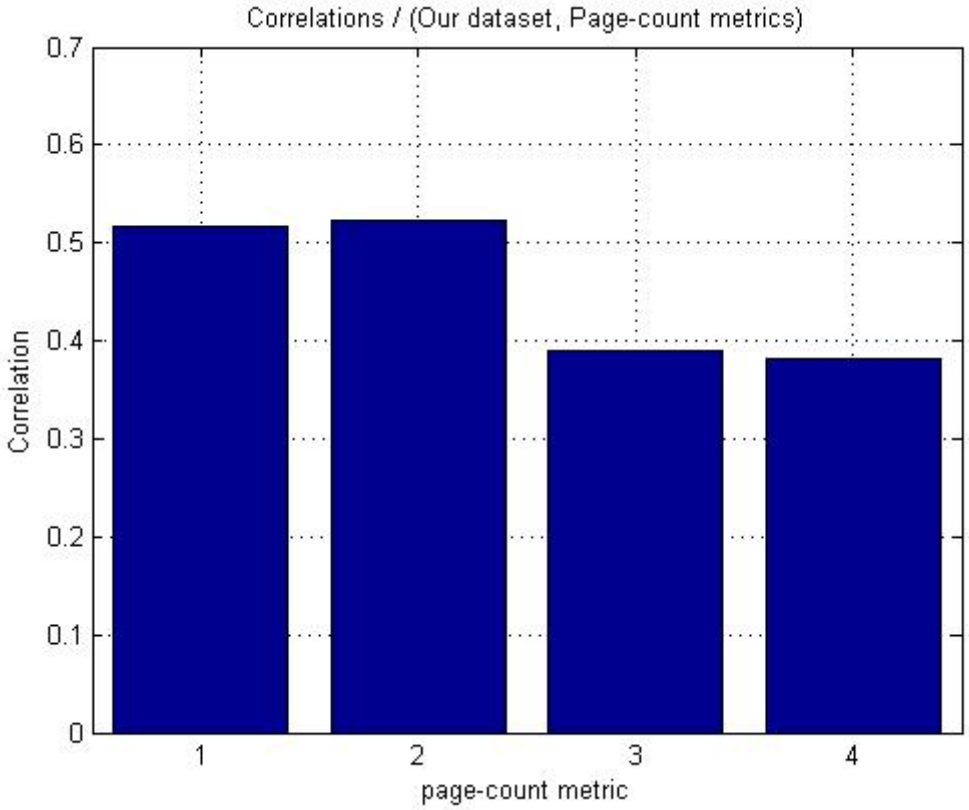

Fig.8 Correlations / (100 URLs, non-translated dataset, page-count metrics, TURBO-filtered contexts)

Both diagrams appear as one! The differences are actually insignificant.– In both diagrams the prevailed metrics are the NGD and MI achieving a correlation score (almost literally in both figures) of 0.51 and 0.53 correspondingly , while the other two ones, the Dice and Jaccard achieve a correlation score of 0.39 and 0.38 correspondingly (again, almost literally in both figures).
Looking the figures we understand that there is not actually a metric which can hit high! Even the NGD and MI cannot even "overcome" the "correlation base".

Now, let's repeat this diagram storm (!) also for the other dataset, the translated. Here comes the first figure:

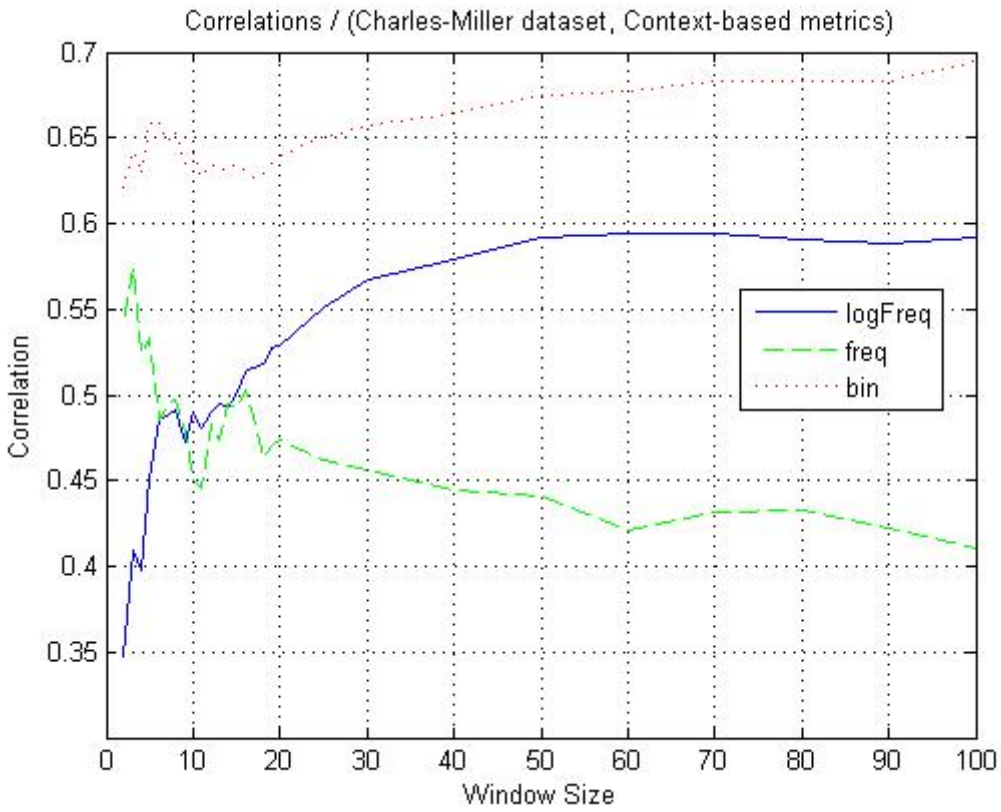

Fig.9 Correlations / (100 URLs, translated dataset, context-based metrics, not TURBO-filtered contexts)

That seems bad! Totally different from the other set; no similarity seems to exist between the two datasets. The metrics appear, actually, out of control! LogFreq appears this time not the prevailed metric at all, while bin metric seems to achieves high scores... For both metrics, bin and logFreq, and in contrast with the previous figures, we notice that the correlation increases as WS increases. Moreover, the freq metric also doesn't look good.– Despite the "bin metric" achieves a high correlation for a big WS, we cannot consider it as trustworthy because, as we said, of 1) the uncontrolness of the figure, and 2) the "unnatural" big WS in which the high score is achieved.

At next figure we try TURBO-filtered contexts:

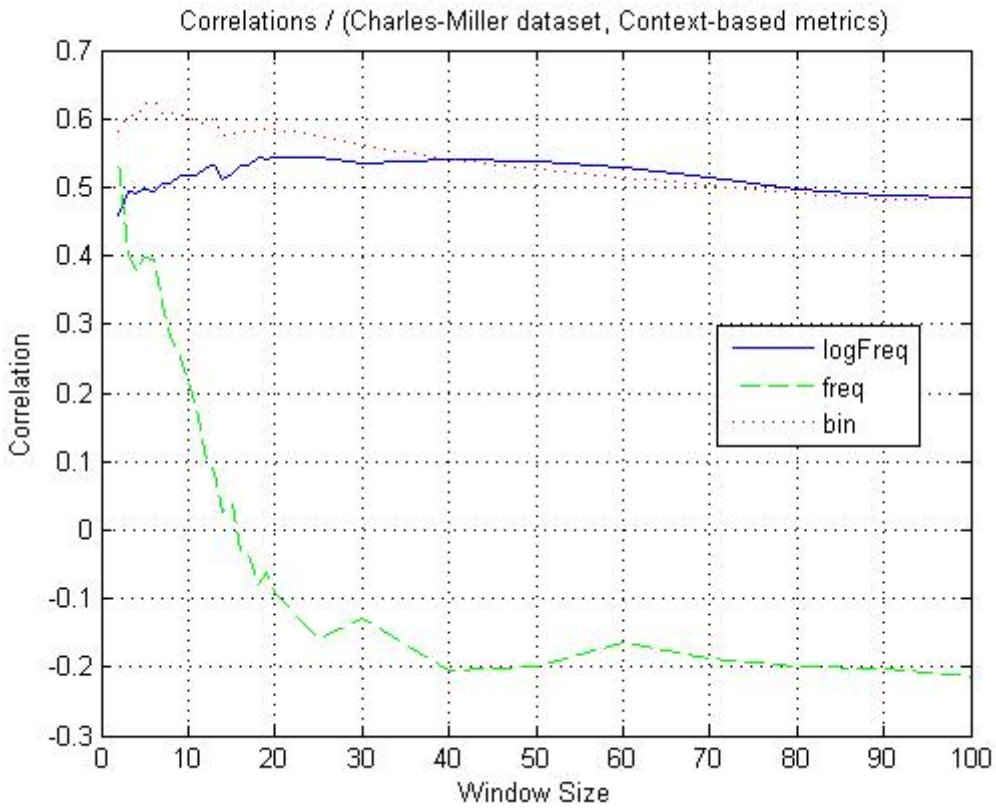

Fig.10 Correlations / (100 URLs, translated dataset, context-based metrics, TURBO-filtered contexts)

Now things appear, in some way, better. The metrics logFreq and bin seem to have calmed down (!); the diagrams are most "soft"; the huge mesh has gone... However, the total correlation has been lessened, and the metrics bin and logFreq appear to be in great distance from the metric freq, which achieves a negative correlation for $WS > 17$.– The conclusion is, again, that the results can not been considered as trustworthy because of the paranormal way they appear.

But let's see how the page-count metrics work (for both, filtered and unfiltered contexts[17]):

Similarly, at next figures, each metric belongs to a number:
1: NGD metric
2: MI metric
3: Dice metric
4: Jaccard metric

[17]TURBO-filtered...

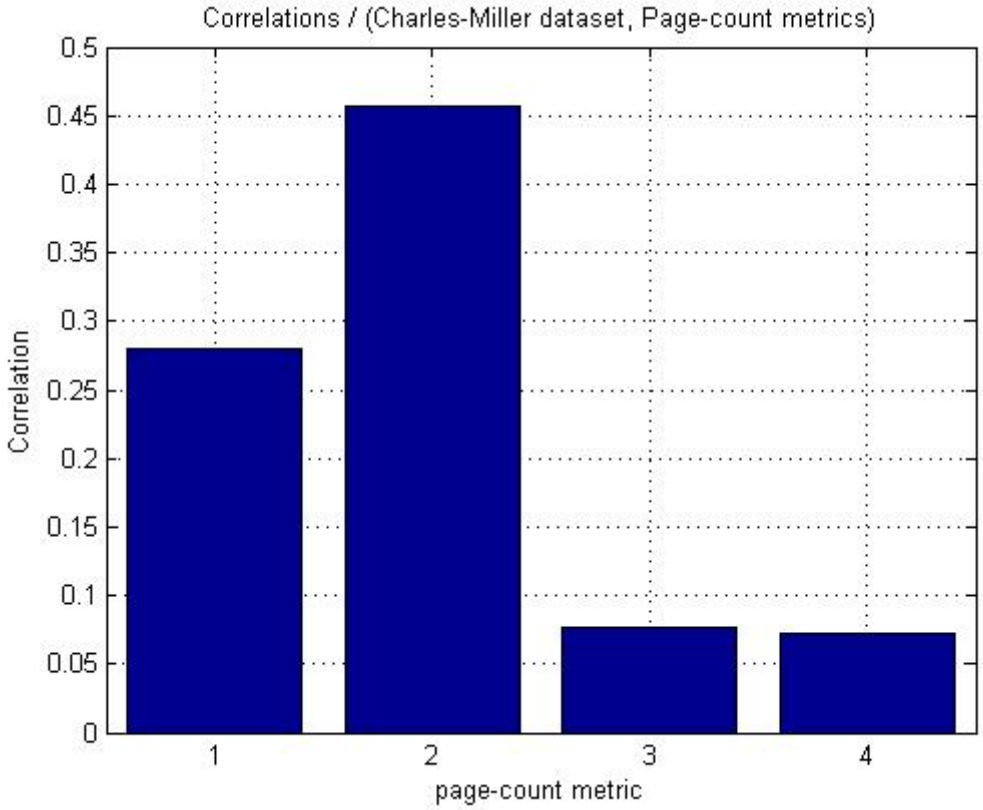

Fig.11 Correlations / (100 URLs, translated dataset, page-count metrics, not TURBO-filtered contexts)

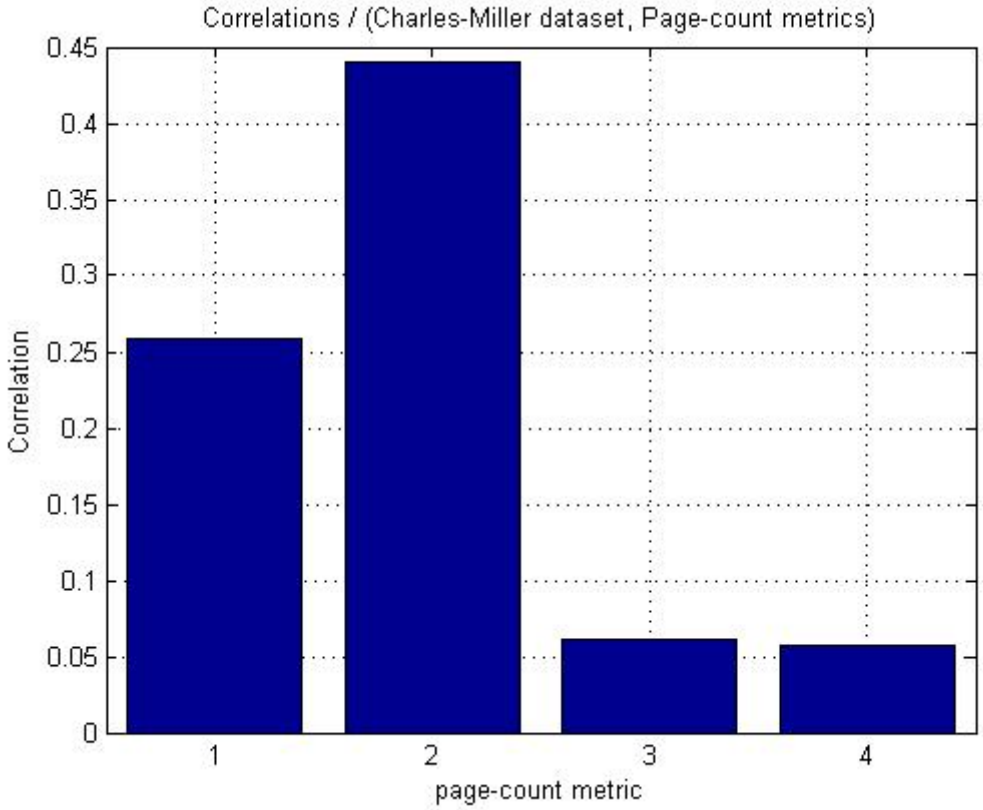

Fig.12 Correlations / (100 URLs, translated dataset, page-count metrics, TURBO-filtered contexts)

Similarly to figures 7 and 8, both diagrams appear, actually, the same, and all metrics are below 0.5 correlation. An obvious difference (comparing these diagrams with the corresponding previous ones) is that metrics NGD and MI are not in a similar correlation level; and this probably something which is added to the non-trustworthiness of these last experiments.

5.2.2 300 URLs

In order to improve our results, we make more experiments on 300 URLs. The idea is that more contexts have more information and therefore, maybe,less noise. The experiments are conducted only for Context-based metrics.

The first figure depicts 'the non-translated dataset and the contexts are all TURBO-filtered. Here it is, the first figure:

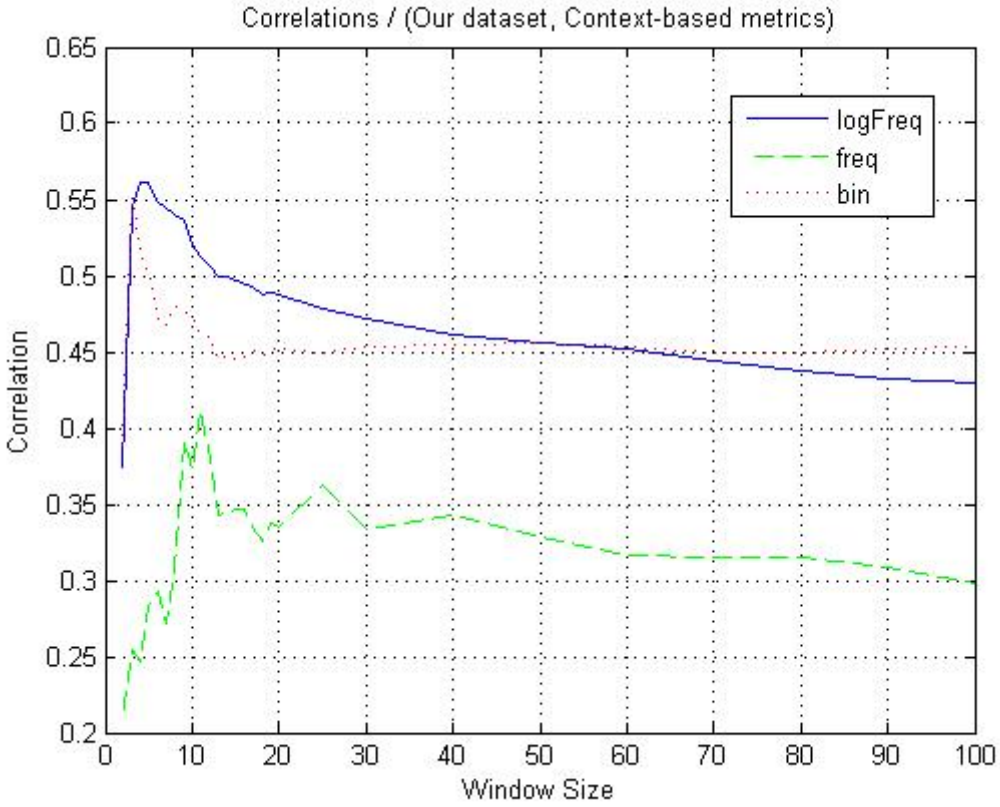

Fig.13 Correlations / (300 URLs, non-translated dataset, context-based metrics, TURBO-filtered contexts)

Comparing this figure to fig.6, we notice that the results are not better at all! Particularly, the correlations have generally lessened; the prevailed metric seems to be logFreq, which achieves a lower correlation score (0.57) than the correspondent of fig.6 – note also that the high-score WS is much smaller, specifically, 7 words smaller; bin metric is still close to logFreq, while freq look like a cardiogram... Additionally we observe (as far as logFreq metric is concerned) that still for small window sizes we have low correlation scores.– The conclusion is that no enhancement has been observed.

Continuing, second figure "depicts" the translated dataset similarly for TURBO-filtered contexts. Here it is:

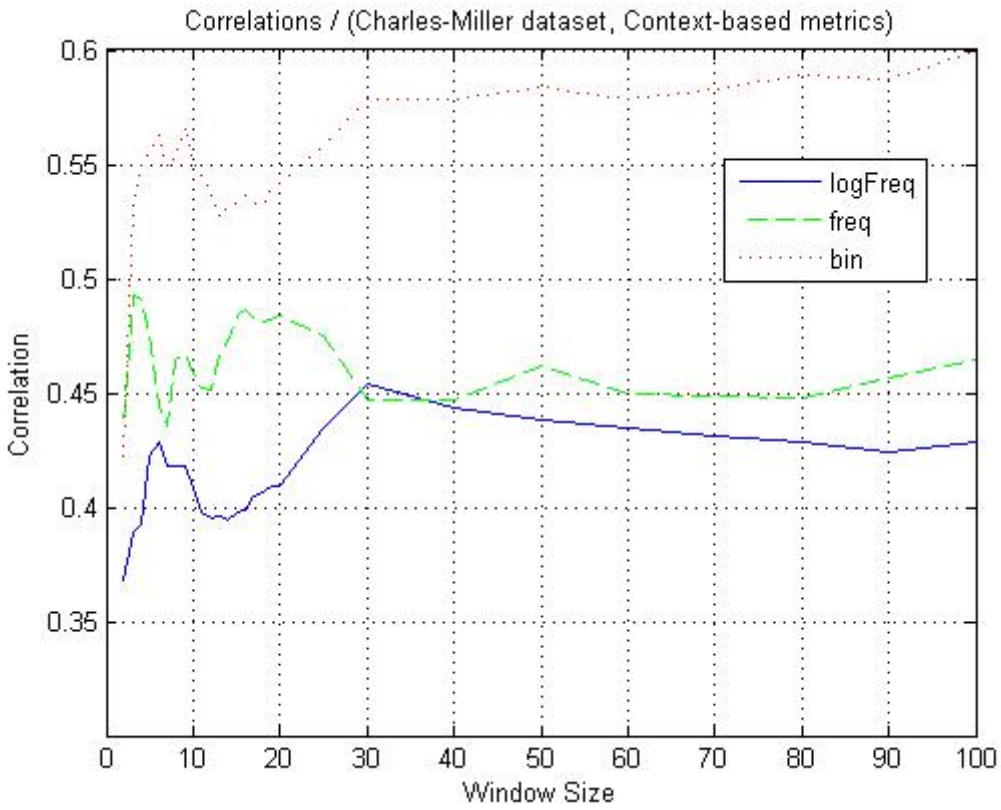

Fig.14 Correlations / (300 URLs, translated dataset, context-based metrics, TURBO-filtered contexts)

Comparing this figure to fig.10, the facts are obvious... the metrics in this diagram are completely disordered! Unfortunately, the results are out of any expectation; the 300 URLs surely didn't help the translated dataset, at all!

5.2.3 More experiments

In order to improve our results, we tried more experimental approaches. Particularly, we tried the following ideas:

1. Four heuristics:

   - $(\$fL - \$fR) < \$aThreshold$, i.e. the frequency difference of a common word (i.e. a word that exists in both, right and left WS) should not be greater than a threshold.
   - $\$fLR < \$aThreshold$, i.e. the frequency of a word, inside both window sizes (left and right together), should not be greater than a threshold.

- $\$ftotal < \$aThreshold$, the total frequency of a word (i.e. inside the whole context) should not be greater than a threshold.
- Top scores grabbing and script re-running on top-score contexts.

2. Stemmer implemantation
3. Stop-word filtering
4. Conversion of ο, η, το to a single το.

Nevertheless, and despite the variety of our ideas, we didn't manage to achieve any better semantic-similarity correlation for both datasets.

Two characteristic diagrams are presented below; both concern the non-translated dataset. The first diagram presents a stemmer application:

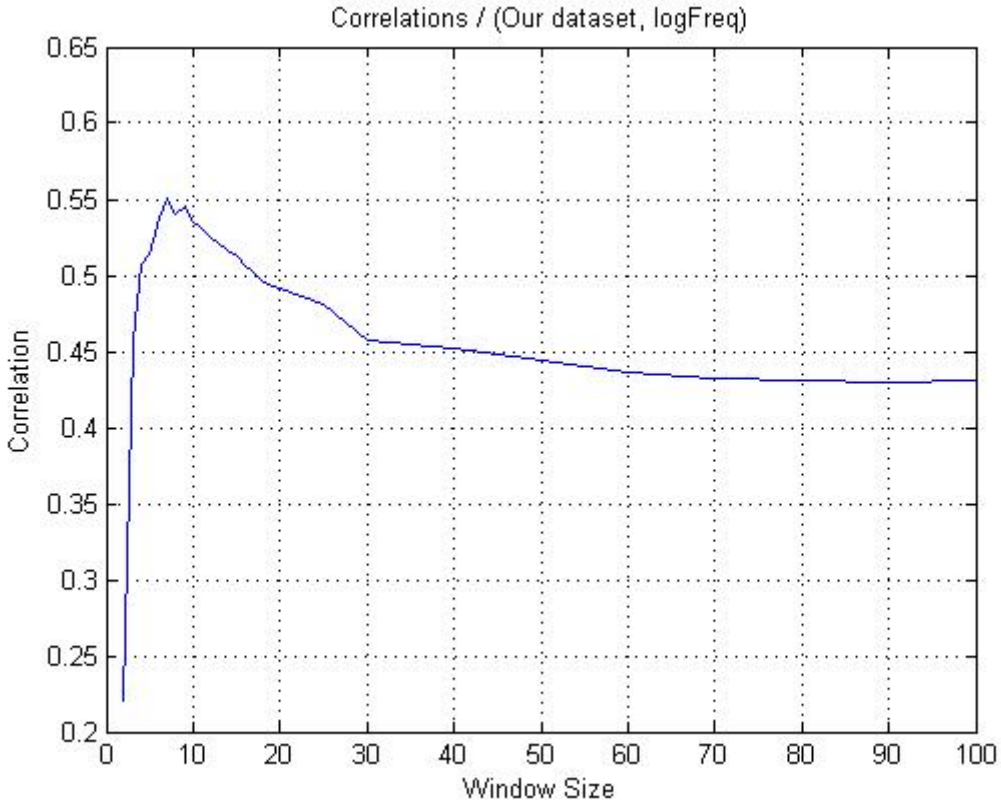

Fig.15 "Stemmer-correlations" / (100 URLs, non-translated dataset, logFreq metric, TURBO-filtered contexts)

This figure presents only the logFreq metric for the non-translated dataset. Comparing to fig.6, we observe that the correlations are generally declined; a correlation peak is achieved for WS 7 (4 words smaller than that of fig.6) which score is 0.55, quite smaller than the 0.68 of fig.6.– As conclusion: no improvement has been observed. But let's see the next figure which is based in contexts without stop-words:

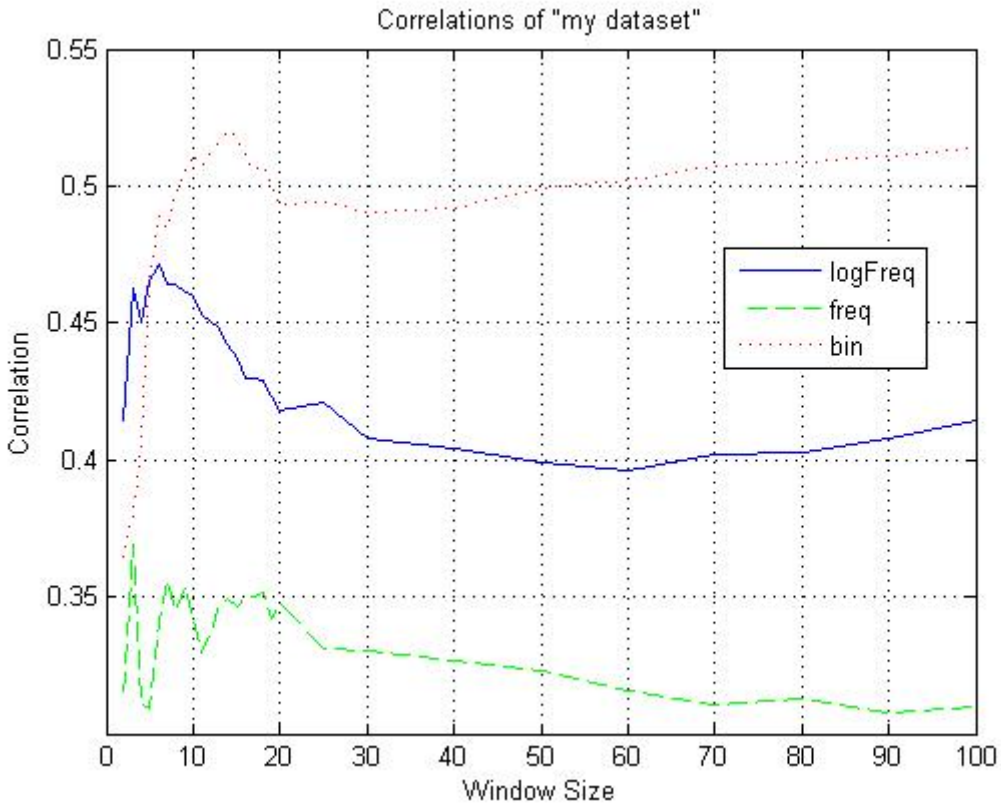

Fig.16 "StopWord-correlations" / (100 URLs, non-translated dataset, context-based metrics, TURBO-filtered contexts)

This figure presents all the context-based metrics for the non-translated dataset. Comparing to fig.6 we conclude that the results are far away of anything, simply, just good! The metrics seem completely disordered; logFreq has been unexpectedly declined, bin metric is pictured here as prevailed, and freq touches the zero-line.– As conclusion: the total figure looks like completely unreasonable.

5.3 English dataset

Since we have done with HTMLs downloading, encoding, clustering and decoding we are now ready to launch our experiments. As we have mentioned in chapter 1, these experiments are based on senses: word groups, context ereas with

common characteristics; the algorithmic procedure in order to calculate the similarities has also been presented in chapter 3.– Now, it's time to examine how senses work...

In order to understand the next figures we present below the correspondences:
1: Baseline result
2: Non-baseline result

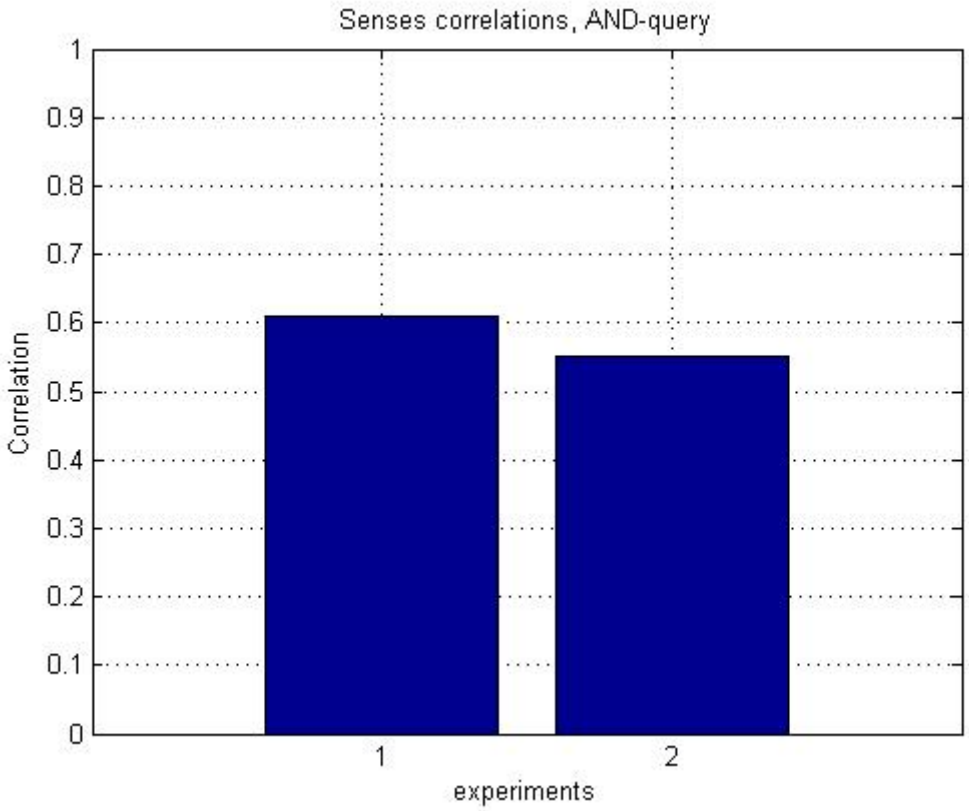

Fig.18 "Senses correlations" / (100 URLs, english dataset, AND-query, binary metric, TURBO-filtered contexts)

This figure presents the "senses correlations" and the metric which is depicted is the context-based binary metric. The left bar shows us the baseline (i.e. the similarities were calculated without senses), while the right bar shows us the main-experiment correlation (i.e. the similarities were calculated with senses).– We observe that, unfortunately, the main-experiment correlation, which achieves the score of 0.56, is lower than the baseline, which achieves the score of 0.61. We remind you that the contexts that the binary metric compares, are the "strong

senses"[18]. Additionally, note that the contexts were downloaded with an AND-query[19]. But let's see the next figure:

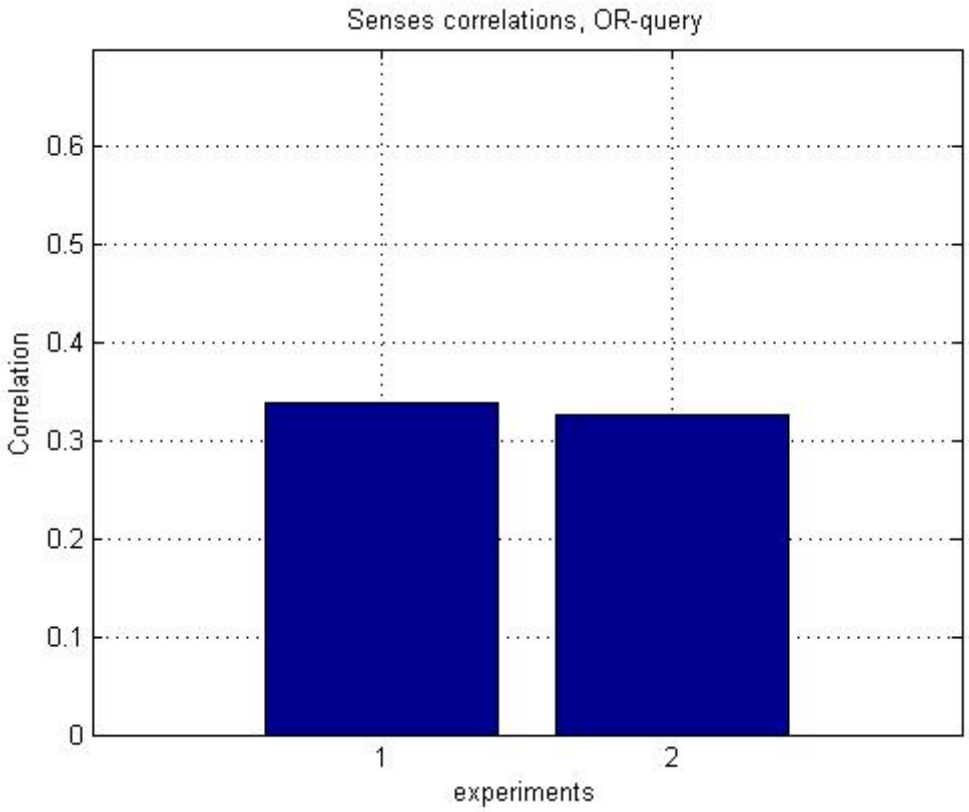

Fig.19 "Senses correlations" / (100 URLs, english dataset, OR-query, binary metric, TURBO-filtered contexts)

This figure also presents the "senses correlation", and the metric which is depicted is the context-based binary metric. Note that the difference is that, this time, we applied an OR-query in order to download our contexts.– Similarly, we observe that the main-experiment correlation is lower that the baseline, and, particularly, both bars are shown to be way down from the 0.5 correlation.– As conclusion: neither this nor the previous query (and for the specific method we used senses) helped in order to improve our results.

[18]One strong sense for each word of the pair

[19]I.e. we commanded the search engine: "look for wordA AND wordB".

6 Discussion and Future Work

Finally, the time of conclusions and future work has come. In this chapter we discuss about the results of our experiments, we try to identify and explain the experimental holes, as also to reveal the technical experimental difficulties we meet. Additionally and lastly, we present our views and opinions about the future work; about what would be generally better, and what research possibilities are "open"!

6.1 Conclusions

As general conclusion we must admit that the experiments didn't produce the "best results". As far as the experiments on Greek datasets are concerned, we have achieved the generally good result of almost 0.7 correlation (fig.6, not translated dataset). The truth is that despite this high correlation score we can't be sure of its "firmness" and "reliability", as binary and frequency metrics are presented far away one from the other (e.g., look for the other case in [3]). However, the "correlation results" between our thesis and paper [3] cannot be considered as important and dependable; the reason: we have to cope with two different languages.

The truth is that all what we said above, may had a strong meaning whether everything was working perfect!

A very important step of the algorithmic procedure is that of filtering, and particularly, HTML-tag filtering. My opinion is that the HTML-tag filter we use in our experiments, in both parts, is not as good as it should be; the reason: E.g. consider 10 HTML files. The filter may filter excellent 4 files; the other 6 will be more or less good, or, in some cases, even bad filtered! However, this is not necessary a problem. In experiments on English dataset the high correlation scores are achieved for a small WS [3]; thus, there is no serious chance of grabbing a HTML-tag. But, what about the experiments on Greek datasets? If we consider as reliable the correlation results, then, may there is a problem, or, maybe, the results not be reliable because of this problem...

Moreover, again for experiments on Greek datasets, we tried to apply a stemmer on contexts. The results, actually, were not totally bad (fig.15); however they could be probably better whether the stemmer was more effective. What we mean is that this Greek stemmer doesn't reveal everytime and for every word the right stem.

Now, comparing the two Greek datasets (translated and non-translated) we observe that the non-translated dataset "behaves" in a more "smooth" or even "logical" way, in contrast with the translated dataset which in most of the figures appears chaotic... This fact make us curious of... what's going on! In this thesis the reason of why is this happening, hasn't been examined.

As far as the experiments on English datasets are concerned, we conducted two experiments (we applied two queries: an OR-query and an AND-query), in which the main-experiment correlations were below the baselines. Particularly, despite both experiments didn't produce good results, we easily observe that (comparing the baselines) the AND-query works better than the OR query.

6.2 Future work

As far as the future work is concerned, first and foremost it is suggested be "bought" a HTML-tag filter. This may be a right "move" in order to improve (even slightly) both general results (i.e. for Greek and English datasets).
Now, particularly, as far as Greek datasets are concerned, we recommend the following ideas. One idea could be this: A new research to be conducted, with new datasets in order to "see" how the new datasets behave in order to compare them with the old ones and be able to make safer conclusions... Moreover, a better subject[20] organization (e.g. as in [1]) could also enhance the correlation, as the human similarity scores will be then more "concentrated"! Additionally, a more effective stemmer will be very helpful.
Continuing to the "senses part" we could recommend, e.g., a senses linear combination! In this thesis we implemented the "rule of strong", i.e. we only apply the metrics between the prevailed, the "strongest" senses of a pair. Furthermore, an other class-impurity criterion could give maybe different results; we use entropy.
Even moreover, we can imagine the following: Smart Web Search Engines that not only do a "semantic-search", they also offer "semantic-choices" to the searcher! Or imagine YouTube-kind or Amazon-kind web sites which do additional searches based on contexts.

[20] Questionnaire subject

6.3 Epilogue

This is our thesis, these are our results, these are our visions! We hope to have set a solid base for the following researchers, and we wish... science next to mankind!

References

[1] Herbert Rubentstein and John B. Goodenough (1965), *Contextual Correlates of Synonymy*, Decision Sciences Laboratory, L.G. Hanscom Field, Bedford, Masschusetts.

[2] George A. Miller and Walter G. Charles (1991), *Contextual Correlates of Semantic Similarity*, Language and Cognitive processes.

[3] Elias Iosif and Alexandros Potamianos (2007), *Unsupervised Semantic Similarity Computation using Web Search Engines*, IEEE/WIC/ACM International Conference on Web Intelligence.

[4] Euripides G.M. Petrakis, Giannis Varelas, Angelos Hliaoutakis, Paraskeui Raftopoulou (2006), *X-Similarity: Computing Semantic Similarity between Concepts from Different Ontologies*, Department of Electronic and Computer Engineering, Technical University of Crete (TUC).

[5] David Yarowsky (1995), *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods*, Department of Computer and Information Science, University of Pennsylvania, USA.

[6] Bollegala D., Matsuo Y., Ishizuka M. (2007), *Measuring Semantic Similarity between Words using Web Search Engines*, Proc. Int. WWW2007 Conf.

[7] Feldman R., Dagan I. (1998), *Mining Text using Keywords Distributions*, Journal of Intelligence Information Systems.

[8] Pargellis A., Fosler-Lussier E., Lee C., Potamianos A., Tsai A. (2004), *Auto-Induced Semantic Classes*, Speech Communication 43, 183-203.

[9] Siu K.-C., Meng H.M. (1999), *Semi-Automatic Acquisition of Domain-Specific Semantic Structures*, Proc. EUROSPEESH.

[10] Iosif E., Tegos A., Pangos A., Fosler-Lussier E., Potamianos A. (2006), *Unsupervised Combination of Metrics for Semantic Class Induction*, Proc. IEEE/ACL Spoken Language Technology Workshop.

[11] Sebastiani F. (2002), *Machine Learning in Automated Text Categorization*, ACM Computing Surveys, 34(1):1 47.

[12] D. Jurafsky, J.H. Martin (2000), book: *Speech and Language Processing*