Masters in Informatics Engineering
Thesis
Final Report

Semantics in Place and Time

João Oliveirinha
jmforte@student.dei.uc.pt

Advisor:
Francisco Câmara Pereira and Ana Oliveira Alves
Date: July 20, 2010

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Abstract

During the last few years, the amount of online descriptive information about places and their dynamics has reached reasonable dimensions for many cities in the world. This enables for a new dimension of understanding space, particularly in which respects to what *exists* there and what *happens* there. Information techniques are needed for extracting the meaning of places that underlies these massive amounts of commonsense and user made sources.

It is presented in this thesis a methodology to automatically label places based on the events occurring near them. To achieve this we use Information Extraction techniques applied to online resources such as Upcoming, Wikipedia and Boston Calendar.

This report approaches the first part of the work that is currently being developed and integrated in the larger context of Semantics and the City.

Keywords: automatic tagging, semantic index, labeling places, event semantics

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, we live in a world that is surrounded by information. This information is everywhere and can be accessed through a desktop PC, a mobile phone or even a TV. With so much information it is impossible to process or search all this data in realistic time. Thus, it becomes clear that it is important to index all this information to facilitate search and processing operations on large-scale information retrieval systems.[5]

Usually the information is received in small discrete blocks that do not relate to each other. These blocks can be homepages, news, events, images, videos or even tweets. In this context, the index needs to applied to each of these blocks instead of to information as a whole, or its source of information. One of the resources available for indexing information that came up with the emergence of Web 2.0 are the *tags*, which consist of annotating bits of information with concepts that do not follow any taxonomy or ontology, they are freely assigned by users. Tagging - the act of adding tags to a resource - is a method that social networks often apply to define resources in a more flexible and variable way, meaning that tags result in unstructured knowledge. This knowledge then becomes composed by a list of concepts that are nothing more than cognitive units of meaning. A cognitive unit of meaning corresponds to an abstract idea or mental symbol that can be represented as one or more words[6].

On the other hand, one of the characteristics of information provided by web resources is geo-reference, meaning that we have an absolute position, such as the pair longitude-latitude. Such powerful information introduces location-aware concepts to information retrieval systems. Taking as an example, there are events being hosted in specific places, advertised on the internet in specific event listing websites or even tweets (that are starting to have geo-point information as a testing service for developers). This type of geo-referenced information introduces a new possibility for more location-aware services, and as a consequence the definition of place could be enhanced[7].

As argued before [8], absolute position such as the pair latitude/longitude is a poor representation of a place because of the different human perspectives and dimensions. From the human perspective places are often associated with meaning, and different people relate to places in different ways. The meaning of place can derive from social conventions, their private or public nature, possibilities for communication, time, and many more. Distinguishing between the concept of place from space, a place is generally a space with something added - social meaning, conventions, cultural understandings about role, function and nature. Thus, a place exists once it has meaning for someone and the perception of this meaning is the main objective of this thesis. It is important to note that the meaning of place here can be a point of interest or a geographic area.

If we now consider that the information has a time stamp associated, we can define a place as a function where time is the variable and the result is a list of concepts that better describe the place. These lists are referred in multiple sections of this report as semantic indexes. Therefore, what we intend to achieve by the end of this research is to describe places in a time window by exploiting the events they hold, and in the process describing the events themselves.

1.1 Motivation

Location-aware systems are now quite important as they allow the user to retrieve better results based on current location. As an example, we can consider the existing applications for the current smartphone market, such as for Android [1] and IPhone [2]. Both have applications that can change the current state of the system based only in current location, like changing the desktop wallpaper or switching the equipment to vibration mode.

However this type of system is programmed to manually change the state of the system by specifying the locations for the change to occur. In this dissertation, the long term research topic addressed is to automate this process, meaning that the user does not need to specify the exact location, but instead he specifies that he

[1] http://en.wikipedia.org/wiki/Android_(operating_system)

[2] http://en.wikipedia.org/wiki/IPhone

wants more volume in his handset when he is in the shopping centers or in loud places.

On the same time, the search engines have also begun to explore the search based on location, sometimes referred to as *Local Search*. A location based query consists of a topic and a reference location, and unlike general web search, a local web search is expected to return documents ranked by their relevance or another variable but the most important aspect is the geographical relevance to the location specified.

There are several issues for developing effective geographic search engines and, as yet, no global location-based search engine has been reported to achieve them [9]. Some of these difficulties can be described as: location ambiguity, lack of geographic information on web pages, language-based and country-dependent addressing styles, multiple locations related to a single web resource and lack of structure in data from multiple information sources.

Search engine companies have started to develop and offer location-based services. However, they are still geographically limited, mostly to the United States, such as Yahoo!Local, Google Maps and MSN Live Local, and have not become as successful and popular as general search engines.

Despite this, plenty of work has been done in improving the capabilities of location-based search engines [10], but it is beyond the scope of this internship to develop them. Instead, the role of this research in this context is more on the side of contributing to the indexing capabilities of such engines in terms of local search than on becoming any alternative form of search per se.

1.2 Objectives

The work proposed for this internship is to design and develop a system capable of extracting a semantic index from various geo-referenced web sources and that describes a place or event in time. As explained in the introduction and motivation section, a semantic index is a small list of concepts that can be interpreted as a tag cloud, that better describes the place being evaluated. The objectives of this internship are to continue the development of the current system of extracting semantic indexes from online resources [11] and improve their creation by taking into account the dynamics of place and time that are associated to these resources.

To achieve this, it was required to build a system that implements a set of functionalities:

- Inclusion of new sources for events.
- A sub-system that removes noisy data.
- Efficient weighting system towards event related texts.
- A methodology for extracts and integrating semantic indexes capable of describing an area or place.
- Improvement of the semantic indexes extracted by adding an enrichment layer.

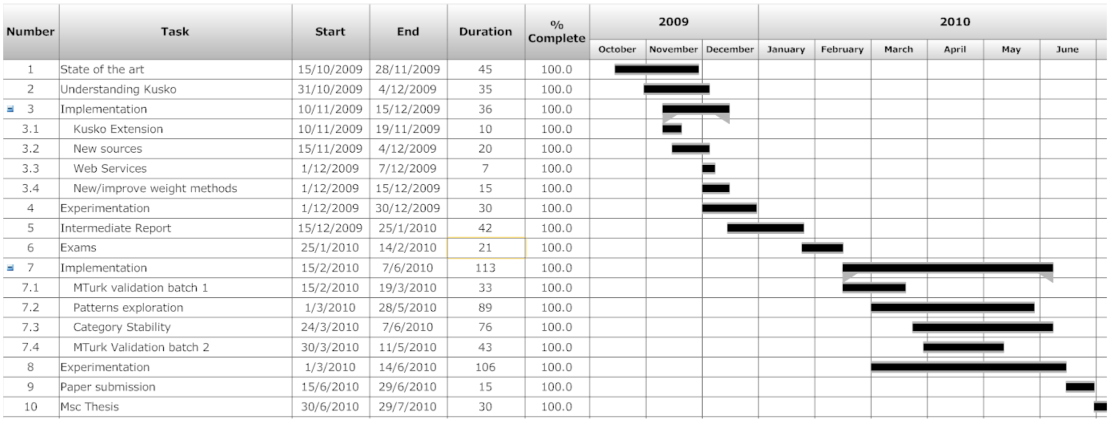
Another objective of this research was to validate the results and see the overall performance of this methodology, as well as if it was possible to extract simple and complex patterns from the flow of events.

1.3 Results

The results of this research dissertation can be found in chapters 4 and 5. In general, we can say that the results obtained are good because this methodology can really extract semantic information from unstructured text, even if the information is not exactly presented on the events descriptions. This is only possibly because of the *Enrichment* stage that is introduced in section 3.3.

On the other hand, if the sources used do not provide the system with sufficient information, the semantic indexes extracted do not provide us with much information about the place. This happens because we deal with dynamic sources, meaning that information submitted to sources come from online users and sometimes the textual event descriptions are only one paragraph or less.

From the work done in those sections, it can be seen that it is also possible to automatic label events and venues using a previous known taxonomy of categories, as well, as the possibility of extracting patterns from the flow of events.



Number	Task	Start	End	Duration	% Complete
1	State of the art	15/10/2009	28/11/2009	45	100.0
2	Understanding Kusko	31/10/2009	4/12/2009	35	100.0
3	Implementation	10/11/2009	15/12/2009	36	100.0
3.1	Kusko Extension	10/11/2009	19/11/2009	10	100.0
3.2	New sources	15/11/2009	4/12/2009	20	100.0
3.3	Web Services	1/12/2009	7/12/2009	7	100.0
3.4	New/improve weight methods	1/12/2009	15/12/2009	15	100.0
4	Experimentation	1/12/2009	30/12/2009	30	100.0
5	Intermediate Report	15/12/2009	25/1/2010	42	100.0
6	Exams	25/1/2010	14/2/2010	21	100.0
7	Implementation	15/2/2010	7/6/2010	113	100.0
7.1	MTurk validation batch 1	15/2/2010	19/3/2010	33	100.0
7.2	Patterns exploration	1/3/2010	28/5/2010	89	100.0
7.3	Category Stability	24/3/2010	7/6/2010	76	100.0
7.4	MTurk Validation batch 2	30/3/2010	11/5/2010	43	100.0
8	Experimentation	1/3/2010	14/6/2010	106	100.0
9	Paper submission	15/6/2010	29/6/2010	15	100.0
10	Msc Thesis	30/6/2010	29/7/2010	30	100.0

Figure 1.1: Gantt chart for work distribution.

1.4 Work Distribution

As can be seen, the figure 1.1 depicts the work distribution for this internship. Since some of the tasks are related, it can be seen that it is relatively easy to parallelize them. This explains why it was possible to start implementation of some features and at the same time reading documentation about *Kusko*, which is described in section 2.5.1.

A larger chart is available in appendix A.

1.5 Outline

Chapter 2 deals with the theoretical background necessary to develop the work here described. It is also presented some document and term similarity measures that were used in some parts of the validation and experiments made to data produced by the system.

In the last two sections of this chapter, it is presented and analysed the related work with this research project.

In the subsequent chapter, we present our methodology for extracting semantic indexes from places and events, along with all the details about the system.

In chapter 4 we present and analyse some examples from events and their respective semantic indexes extracted from the city of Boston, as well as the use of these events as a way to compute and classify the semantic indexes of the venues. It is also presented another experiment that acts as a proof of concept for the extraction of patterns of events by the exploration of their dynamics and similarity between them.

The chapter 5 is where we focus on the validation of this research project and some statistical results. This chapter is divided into two sections, where one is based on volunteers to validate the methodology and the other is the automatic validation of the system by the use of a set of algorithms.

Finally, in the last chapter of this thesis, we present some conclusions and discuss some ideas for improvement of our system and future work.

Chapter 2

State of the Art

2.1 Information Extraction

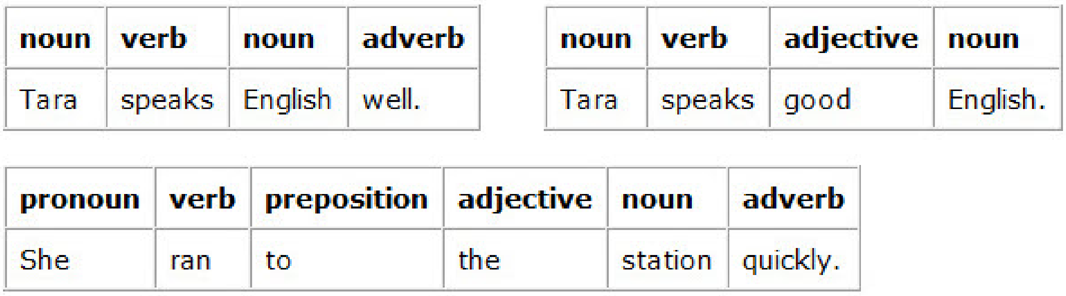
A subtask of Information Retrieval (IR), Information Extraction has as main goal to automatically extract structured information from unstructured machine-readable documents[12]. To accomplish this, Natural Language Processing (NLP) tools are commonly applied.

The result of this process is a categorized, contextually and semantically well-defined data that can be used to allow logical reasoning and inferences based on the relevant content of the document.

Linguistic analysis of text normally proceeds in a layered fashion where the sub-tasks are well defined. In the case of Information Extraction we find five stages:

- Content Noise Removal, which consists of the removal of data that is not important. For example, XML, HTML tags that are retrieved in webpages. There are some NLP tools and frameworks[13][14] that already simplify this work by doing the screen scraping - act of retrieving the text from a web page and stripping the HTML tags while processing it.

- Named Entity Recognition, also referred as Entity Identification, is one of the most important sub-tasks because it tries to identify proper entity names like personal names or organizations, places and temporal expressions. NER algorithms, unlike some noun phrase extractors, tend to disregard part of speech



noun	verb	noun	adverb
Tara	speaks	English	well.

noun	verb	adjective	noun
Tara	speaks	good	English.

pronoun	verb	preposition	adjective	noun	adverb
She	ran	to	the	station	quickly.

Figure 2.1: Part-of-Speech Tagging example.

information and work directly with raw tokens like ”Mr.” or ”Inc.”. Entity Extraction algorithms have the ability to recognize previously unknown entities and are implemented by using recognition and classification rules that are triggered by distinctive features associated with positive and negative examples On the other hand, some entities may pass without being recognized because of the nonexistence of an entity database or ontology like CrunchBase[15] for instance.

- Co-reference resolution stage is where the detection of co-reference and links between text entities takes place. This stage detects whether multiple expressions in one or more sentences refer to the same reference. The sentences “João is working in his research thesis” or “Informatics Engineering Department is known as DEI” are some examples because both sentences use pairs of words to describe the same entity, (João/his) and (DEI/Informatics Engineering Departament), respectively.

- Terminology Extraction, or Concept Extraction, consists of finding relevant terms for the given corpus of documents. The most common approaches of term extraction use Part of Speech tagging and/or Noun Phrase Chunking processors. POS taggers are used to tag each word of the sentence with its grammatical class like identifying nouns, verbs, adjectives, adverbs, etc.[16] (see figure 2.1) This stage, as we will see in section 2.5.1, is a very useful starting point for semantic similarity and knowledge management.

- Finally, Relation Extraction is the final stage that is responsible for identifying relations between entities. For example, ”Barack Obama is the president of the USA”. There are some tools that simply extract these relations from the document and others that get their relations based on other sources as we will see in section 2.5.3.

2.2 Vector-Space Model

Representing a document in a way that computers can understand has some difficulties because of the nature associated to both resources. In fact, it was in the late 1960s that a model capable of representing documents was first used on a project called *Smart Information Retrieval System.*[17] The Vector space model, or term vector model, is an algebraic model of representing text documents as vectors of identifiers that could be indexing terms.

$$D = (t_1, t_2, ..., t_i) \tag{2.1}$$

As expression 2.1 shows, each element represents a dimension that corresponds to a separate term that is present in the document, where t_i represents the ith concept/term in the document.

Normally, also associated with each term is a value that represents the weight of the corresponding term in the original document.

Thus, expression 2.1 could be also defined as expression 2.2, where w_i is the ith weight associated to the ith concept of the document.

$$D = (t_1, w_1; t_2, w_2; ...; t_i, w_i) \tag{2.2}$$

At first, this value can be calculated by just using a binary scheme, where the value is 1 if the word exists or 0 otherwise. From here, calculating similarity between a Document and a Query can be achieved by evaluating the expression 2.3 which results in the scalar product of the two documents. This expression is also known as *cosine vector similarity* and in fact what is being calculated is the angle between the two vectors by doing the dot product between the weights of each concept of the document (w_{qi}) and the weights of each concept of the query (w_{di}).

$$similarity(Q, D) = \sum_{i=1}^{l} w_{qi}.w_{di} \tag{2.3}$$

In practice, it has being proven[17] useful to provide a discrimination among the terms assigned for content representation, meaning that terms with weight closer to 0 would be the least important while closer to 1 would be the most important. So, in some circumstances, it may help to normalize the vectors. This results in a new formula 2.4.

$$similarity(Q, D) = \frac{\sum_{i=1}^{l} w_{qi}.w_{di}}{\sqrt{\sum_{i=1}^{l}(w_{qi})^2 . \sum_{i=1}^{l}(w_{di})^2}} \tag{2.4}$$

With this new expression the weight of a single term depends on the weight of other terms that represent the document.

The normalization of vectors by itself will not help much to improve the results because we are still dealing with a binary scheme. So, if we replace the binary scheme by the term frequency of that word we would now have a better value that can really represent the value weight of a specific term in the document.

This value can be calculated as demonstrated in expression 2.5.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.5}$$

But this approach also introduces a new problem because the term frequency alone cannot ensure acceptable retrieval performance, particularly, when the high frequency terms are not concentrated in a few particular documents. It makes more sense if terms that have a higher term frequency in several documents should have a small weight value as a result of being a common term in the whole collection of documents.

As a result a new factor was introduced: the *Inverse Document Frequency* which represents the number of times a word was present in all documents.

$$idf_i = \log \frac{N}{n} \tag{2.6}$$

The value of N in the IDF expression2.6 represents the number of documents in the corpus/collection and n is the number of documents where the term t_i is present.

Thus, the final expression for the TF-IDF weighting function is:

$$(tfidf)_{i,j} = tf_{i,j} \times idf_i \tag{2.7}$$

Vector models usually work very well for representing documents when mixed with TF-IDF, as a weighting system. Salton[17] confirmed that a normalized TF-IDF was the best system to represent text documents when he realized severals experimentations with other weighting systems. Some of the systems that were tested were variations of TF-IDF, separate and binary TF and IDF.

However, there are some limitations inherent to this model. Long documents tend to be poorly represented because they have small similarity values as a consequence of large dimensionality and small scalar product. Secondly, as we are working with vectors of terms, to achieve a positive match it is necessary that keywords precisely match document terms, and documents with similar topic but using different words won't be similar, thus resulting in false negative match. Finally, the order in which the terms are in the document is not preserved when represented in the vector model.

2.3 Document Similarity Measures

In this section we present some of the best known document similarity measures applied to documents. This type of information is important because it can tell us how similar two events are.

In the last section (2.2) we have already presented the cosine similarity measure as a function that is capable of finding the similarity between a document and a query. This measure is one of the best, as argued by [17], because it takes into account the weight of the concepts and tries to find the angle that is made from the two documents.

Other approaches exist that try to compute the similarity by using a binary scheme. One of these cases is the *Jaccard* similarity measure, that is represented by the equation 2.8. It computes the similarity between two documents by calculating the intersection and union of the two semantic indexes and consequently returning their ratio by diving the two values.

$$J(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{2.8}$$

By doing this, the information about the *term frequency* and *inverse document frequency* is lost and the only aspect that matters is if the concepts exist in the semantic index.

Another function that does not work with the *tf-idf* value of the concepts and is related to the *jaccard* similarity is the *overlap* function. It is represented by equation 2.9.

$$O(A,B) = \frac{|A \bigcap B|}{min(|A|,|B|)} \tag{2.9}$$

This function is very similar to the *jaccard* function, but instead of computing the intersection of the semantic indexes from the two documents, it computes the minimum cardinality of both sets and returns the value of the ratio.

The final similarity function discussed here is the *tanimoto* similarity and is represented by the equation 2.10. This function is an extended version of the *cosine* similarity function as it only uses as the denominator an additional two values that represents the norm of the semantic indexes.

$$T(A,B) = \frac{A.B}{||A||^2 + ||B||^2 - A.B} \tag{2.10}$$

2.4 Concept Similarity Measures

Another type of similarity measures that exists is the one that evaluates the semantic similarity between two terms. In this section we discuss some of these functions, where the first set of functions are semantic measures that are made using as a resource the *Wordnet Ontology*. These functions could lead to a problem because of their use of Wordnet, which could mean that some of the terms we need to compare may not be present in the ontology. For this reason we also present two other similarity measures that do not have any limitation regarding any taxonomy (google distance and edit distance).

The first and one that uses *Wordnet* is the *Path Distance Similarity*[18]. This semantic similarity function returns a score based on the shortest path that connects the senses in the is-a (hypernym/hyponym) taxonomy. The score is in the range of 0 to 1, where 1 means the concepts are similar (two lemmas for the same synset) except in those cases where a path cannot be found (will only be true for verbs as there are many distinct verb taxonomies), therefore it does not belongs to the function domain.

Another measure is the *Leacock Chodorow Similarity*[19] and it returns a score denoting how similar two word senses are based on the shortest path that connects the senses, as in *path distance*, and the maximum depth of the taxonomy in which the senses occur. The relationship is given equation 2.11 where p is the shortest path length and d is the taxonomy depth.

$$lch(p, d) = -log(p/2d) \tag{2.11}$$

Wu-Palmer Similarity[18] is another function that denotes how similar two word senses are, based on the depth of the two senses in the taxonomy and of their Least Common Subsumer.

The *Information Content* of a concept is the specificity of that concept and is defined as the negative of the log likelihood, $-logp(c)$, where $p(c)$ is the probability of encountering such concept. For example, 'money' has a less information content than 'nickel' as the probability of encountering the concept, p(Money) is much greater than encountering the probability of p(Nickel) in a given corpus. IC is already calculated for those senses present in WordNet (e.g.Highway, Registry).

The LCS of two concepts is the most specific concept that is an ancestor of both, and does not necessarily feature in the shortest path connecting the two senses, as it is by definition the common ancestor deepest in the taxonomy. Typically, however, it will be so.[20] Where multiple candidates for the LCS exist, that whose shortest path to the root node is the longest will be selected. Where the LCS has multiple paths to the root, the longer path is used for the purposes of the calculation.

Resnik Similarity[20][19] is the base distance function for the next that follow. This function returns a score denoting how similar two word senses are, based on the Information Content (IC) of the Least Common Subsumer.

Jiang-Conrath Similarity[20] is based in the *Resnik Similarity* because it returns a score based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node) and that of the two input Synsets. The relationship can be viewed in equation 2.12.

$$jcn(s1, s2) = \frac{1}{IC(s1) + IC(s2) - 2 \times IC(lcs)} \tag{2.12}$$

Lin Similarity[20] uses the same idea as the *Jiang-Conrath* but the equation is a little different2.13.

$$lin(s1, s2) = \frac{2 \times IC(lcs)}{IC(s1) + IC(s2)} \tag{2.13}$$

One function that does not require the use of *Wordnet* to compute the similarity between two terms is the *Google Distance*[21]. This function is presented by equation 2.14 and is based in a very simple principle.

$$G(x, y) = \frac{max(log(f(x)), log(f(y))) - log(f(x, y))}{log(M) - min(log(f(x)), log(f(y)))} \tag{2.14}$$

The function $f(x)$ represents a search of the term x and the return result is the total number of pages found by *Google*. The same applies to the call $f(x, y)$ but this time it will search by the concatenation of the two terms separated by a space. The last variable is the constant M that represents the total number of pages that *Google* has indexed in their databases. It is now quite simple to see that what this function does is computing the similarity between the terms by comparing the number of *hits* of each term and the number of *hits* of the concatenation of these terms. The principle of these functions is that if two terms are very close semantically then the probability of the number of hits of each term being close to the number of hits when we search the two terms is very high. The name of this similarity function has the name *Google* but it can be used with any search engine. The only requirement is to have an idea of the number of pages indexed by that search engine.

The last function used as a similarity measure is the edit distance, also know as the *Levenshtein distance*[22]. This function receives two strings and computes the number of minimum number of actions and cost that is required to transform one string into another by removing, modifying or adding a new character. It is easy to see that this function does not take into account any semantics of the terms used. But regardless of this aspect some researchers have used it as a similarity function for their terms and concluded that they have a good performance by if integrated with other semantic functions[23].

2.5 Concept Extraction

In this section, we present and explain three tools that use information extraction techniques to extract a semantic index list from text documents.

2.5.1 Kusco

Kusco[24] is a tool developed by Ana Alves, a Phd. student and co-supervisor of this internship, and its main objective is to extract a ranked list of concepts given a set of textual descriptions. To be able to extract the semantic index, Kusko applies a process that includes Part-of-Speech tagging, Noun Phrase chunking and Named Entity Recognition (NER) using available NLP tools.

This results in a process layered by stages where sentences are broken into words that are tagged by Part-of-Speech taggers as explained in section 2.1. In the next step, Noun Phrase chunking is made typically by partial parsers and go beyond part-of-speech tagging to extract clusters of words that represent people or objects. They tend to concentrate on identifying base noun phrases, which consist of a head noun, i.e., the main noun in the phrase, and its left modifiers, i.e, determiners and adjectives occurring just to the left of it. In parallel, Kusco uses Named Entity Recognition to identify proper names and may also classify these proper names as to whether they designate people, places, companies, organizations, and the like. Once one term could be identified at the same time as an Named Entity and as a Noun Phrase, Integration and Contextualization of this information is applied using WordNet and Wikipedia as Common Sense resources. Thus it is also possible to identify synonyms between terms, as WordNet is structured in families of words having the same meaning, or “synsets”[25]. Wikipedia is also used as a dynamic resource that contains a lot of Named Entity related articles, where WordNet is not complete. These nouns are contextualized on WordNet and thus can be thought not only as a word but more cognitively as a concept, once WordNet contains more relations between words besides synonym (e.g. part_of, hypernym, hyponym, etc.). Given that each word present in WordNet may have different meanings associated, its most frequent sense is selected to contextualize a given term. For example, the term “wine” has two meanings in WordNet: ”fermented juice (of grapes especially)” or ”a red as dark as red wine”; being the first meaning the most frequent used considering statistics from WordNet annotated corpus.

When using data from different sources, integration of information is imperative to avoid duplicates. To solve this problem, Kusco treats differently common nouns

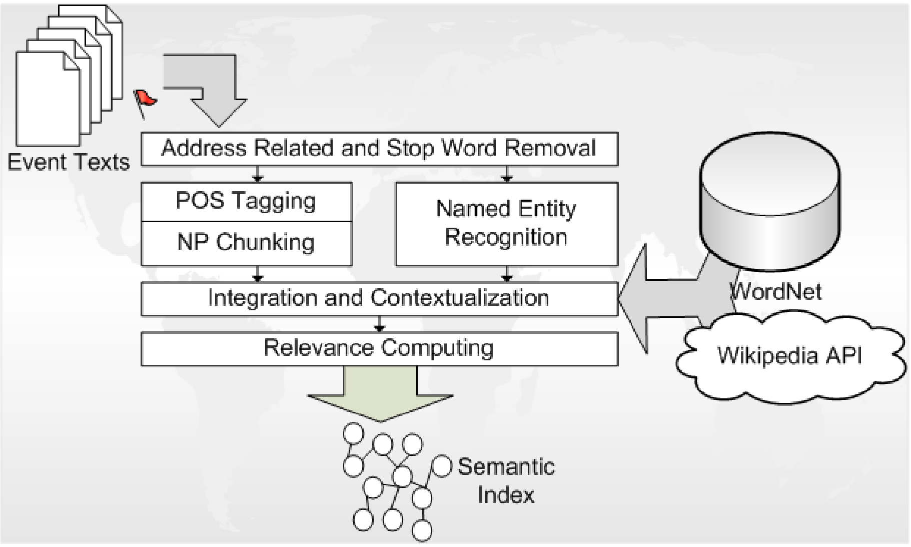


Figure 2.2: Kusco Architecture.

(generally denoting concepts) from proper nouns (generally Named Entities found). Although Kusco uses WordNet to find synonyms in the first group, it does not have a list of all possible entities in the world to match words from the second group. Instead, Kusco takes advantage of the relatively mature field of String metrics to find the distance between strings using an open-source available library with different algorithms implementations[26]. On completion of these subtasks, for each document, Kusco ranks the concept with Term Frequency (TF) value in order to extract the most relevant used terms in the document.

2.5.2 Semantic Hacker

Semantic Hacker[1][27] is another Information Extraction software, developed by TextWise that in addition to extracting a list of concepts is also able to categorize the document content and find similar web content.

To calculate all of this, TextWise developers uses a variation of the Vector-Space Model that they have created called Trainable Semantic Vectors (TSV). TSV is used to generate a semantic index that they call Semantic Signatures that consists of a weighted vector of typically thousands of concepts, which they refer to as *semantic dimensions*. These semantic dimensions are a result of a one-time supervised training process from an appropriate classification schema for the domain and can be labels that represent categories extracted from de Open Directory Project[28].

In addition, they do not require a manual construction or maintenance of ontologies. Instead TSV automatically generates its own semantic dictionary during training that contains the vocabulary known to be relevant to the application domain. See figures 2.3 and 2.4.

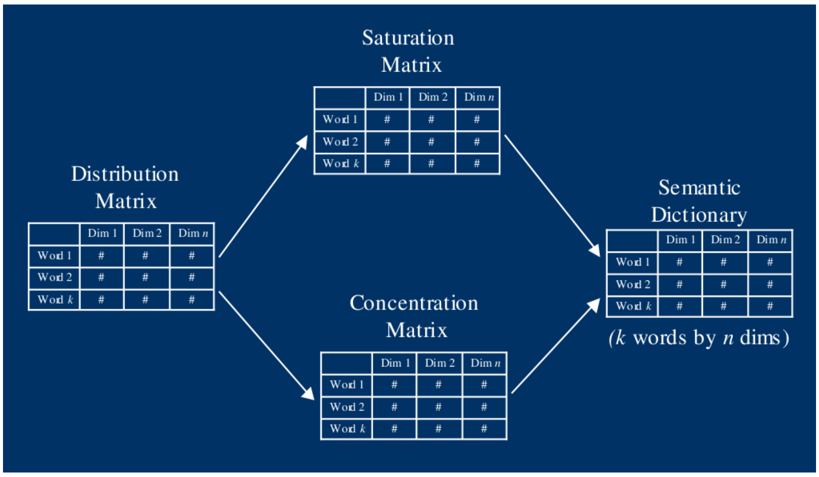


Figure 2.3: Semantic Hacker Dictionary Model[1].

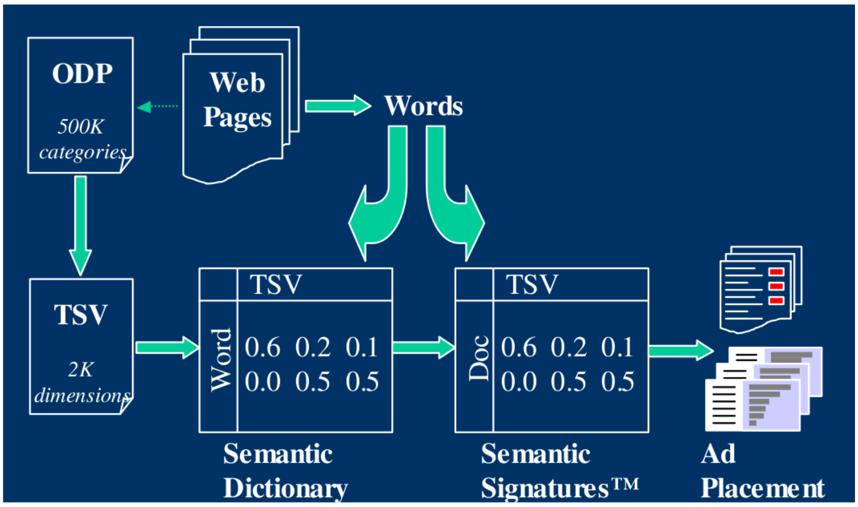


Figure 2.4: SemanticHacker TSV Model[1].

So, for the construction of a semantic signature for a text they rapidly calculate a mathematical combination of all the semantic vectors of the vocabulary contained in the text.

Finally, to compute the relevant content they just compute a match score based on the positions of the vectors in the n-dimensional Euclidean semantic space.

2.5.3 OpenCalais

OpenCalais[2][29] is a software capable of extracting entities, facts and events from unstructured text. Since OpenCalais is a close source product from Reuters, there does not exist much information on how they achieve the results.

It is manly used for automatic tagging of web blog posts.

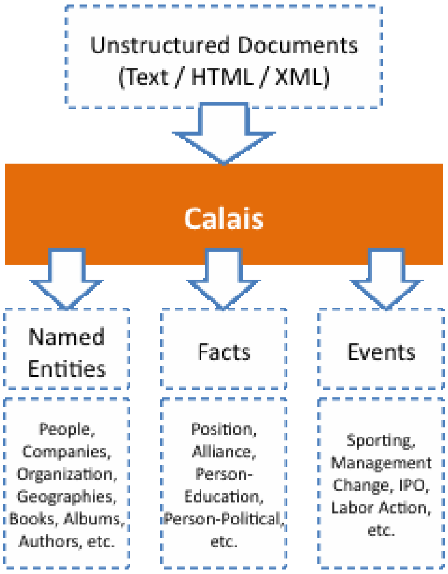

Unstructured Documents
(Text / HTML / XML)
Calais
Named Entities
Facts
Events
People, Companies, Organization, Geographies, Books, Albums, Authors, etc.
Position, Alliance, Person-Education, Person-Political, etc.
Sporting, Management Change, IPO, Labor Action, etc.

Figure 2.5: OpenCalais architecture[2].

Here follows an example of the results that opencalais returns.

Text

George Bush was the President of the United States of America until 2009. Barack Obama is the new President of the United States now.

Entities

Person: Barack Obama (0.29)
Country:United States of America (0.43)
Person: George Bush (0.29)

Topics

Topic: Politics

Relations
PersonPoliticalPast:
Person: George Bush
Position: President
PersonPolitical
Person: Barack Obama
Position: President of the United States

2.6 Semantics and Events

Lemmens and Deng [3] argue that Web 2.0 and Semantic Web have complementary characteristics, and so they suggested an iterative approach of integrating Web 2.0 tags with Ontologies.

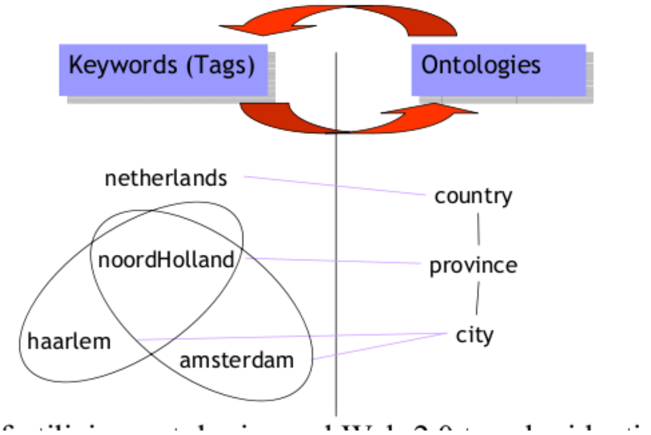


Figure 2.6: Web 2.0 and Ontology[3].

This approach could be used as a semi-automatic tagging process and in fact they explore the right ideas by trying to share the formal soundness of ontologies with the informal perspective of social networks which does not follow any hierarchical structure. However it is almost impossible to implement this system as the main choice points have to be made manually, and for each new POI/category. If we now consider the fact that this type of information is very dynamic, particularly when depending on Web 2.0 social networks, it would demand a set of constant up-to-date resources. In addition to this limitation, they also assume that users have the basic knowledge of semantic standards to make the corresponding match between ontology concepts and tags, which seems away from reality for the current days.

In 2007, Rattenbury et al [4] developed a way to detect events from the Flickr[1] photo Web Service. The idea behind it was to exploit the regularities on the tags assigned to the photos in which regards to time and space of different scales, so when several tags are found within the same small region/place, they become an indicator of event of a meaningful place (See figure 2.7). Then, the reverse process is possible, that of search for the tag clouds that correlate with that specific time and space. They do not, however, make use of any enrichment from external sources, which could add more objective and semantic information to their results. Furthermore, their approach is limited to the specific scenarios of Web 2.0 platforms that carry significant geographical reference information.

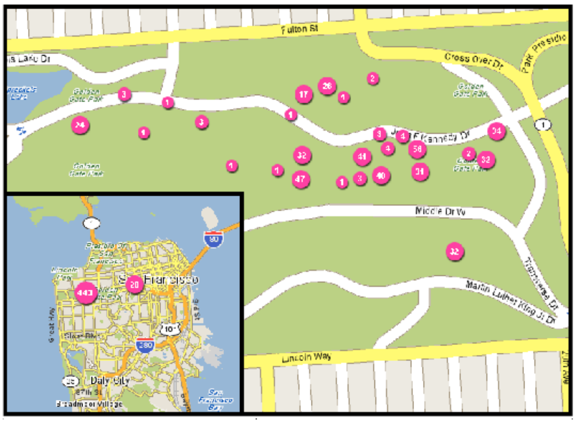


Figure 2.7: Flickr Tags 2[4].

Similar approaches were also made towards analysing Flickr tags by applying ad-hoc approaches to determine ”important” tags within a given region of time [30] or space [31] by exploiting the inter-tag frequencies. However, no determination of the properties or semantics of specific tags was provided [4].

In the Web-a-Where project, Amitay et al [32] tries to link web pages to geographical locations to which they are related. In addition they also also assign to each page a geographic focus that is retrieved by the content the page discusses as a whole. Furthermore, their “tag enrichment” process consists of finding place entities that show potential for geo-referencing, and then applying a disambiguation taxonomy (e.g. “MA” with “Massachusetts” or “Haifa” with “Haifa/Israel/Asia”).

[1] http://www.flickr.com

The results seem to be good, however the authors do not explore the idea other than using explicit geographical references. An extension to this project could be added so it was capable of detecting places using other patterns like Rattenbury et al exploited, and thus without introducing the limitation of explicit geographic content. In fact, Serdyukov et al [33] recently exploited this behaviour by developing a system where pictures are placed in the map given a vector of tags associated to the image.

Chapter 3

Approaches

3.1 Sources

The sources of information used in the system are determinant in this methodology. All the sources were selected with only three characteristics in mind: geo-referenced information, time based information and dynamic information.

First, with geo-referenced information we can know in a more simple way which events are held in a place and pin-point them in a map. Second, time based information gives us the possibility of improving our system by adding a feature of describing a place as a function where time is the variable. However, the most important feature from these sources is the update rate of the information and the fact that this information does not follow any pattern regarding a taxonomy or other type of structure This is caused by the fact that most information is added by communities of web-users.

The sources already used and integrated in this project are:

- Yahoo Upcoming is a web site that exposes a advertiser service for events, and is self-managed by the web community.
- Boston Calendar: This is one web service similar to Yahoo Upcoming but with much more information for Boston, which is one of our targets. Unlike the other sources this does not have an API[1] for extracting information. Thus,

[1] `http://en.wikipedia.com/wiki/API`

the only way was to use screen-scraping which consists in downloading all the web pages and using regular expressions to extract the targeted information.

- Zvents[2] is another web site that provides a service similar to Boston Calendar. In fact, Boston Calendar uses Zvents as a source for themselves. The best advantage is the RESTfull Web Service they expose for better access to their information.

It is important to address that all of these resources are rich in information about various types in events (ie, music events, sport events, lecture events) and the most important information extracted from the sources are: name of the event and venue, location, description, categories and date.

3.2 Concept Extraction

Using this methodology we extract the semantic indexes by applying 4 stages. The first stage is where we retrieve the information in the various sources by using an application developed using the *Python* language.

This process is running using several parallel threads depending on the source, the method used for retrieving information (API or screen scrapping) and the limitations of the service. We have also developed and validated a series of regular of expressions capable of identifying noise related to valid and complete HTML or XML tags in order to strip them from the content.

All the event and venue information is added to a *Postgres* database that will be later processed by a scheduled process, also developed in Python. The schema used for the database is specified in the ER Model presented in figure 3.1 and was developed by a team working in several projects that integrates into a larger system. So, for this research project only the semantic entities are used which are translated in all entities except naics_categories and naics_sectors.

In the next stage, the scheduled process is responsible for feeding Kusco (section 2.5.1) with event information, so we can retrieve a list of ranked concepts. As a second layer of noise removal, we can also provide to Kusko a list of stop words that we do not want to be present in the semantic index extracted. This could act as a dynamic list where it is possible to remove concepts common to the place (eg.

[2] `http://www.zvents.com`

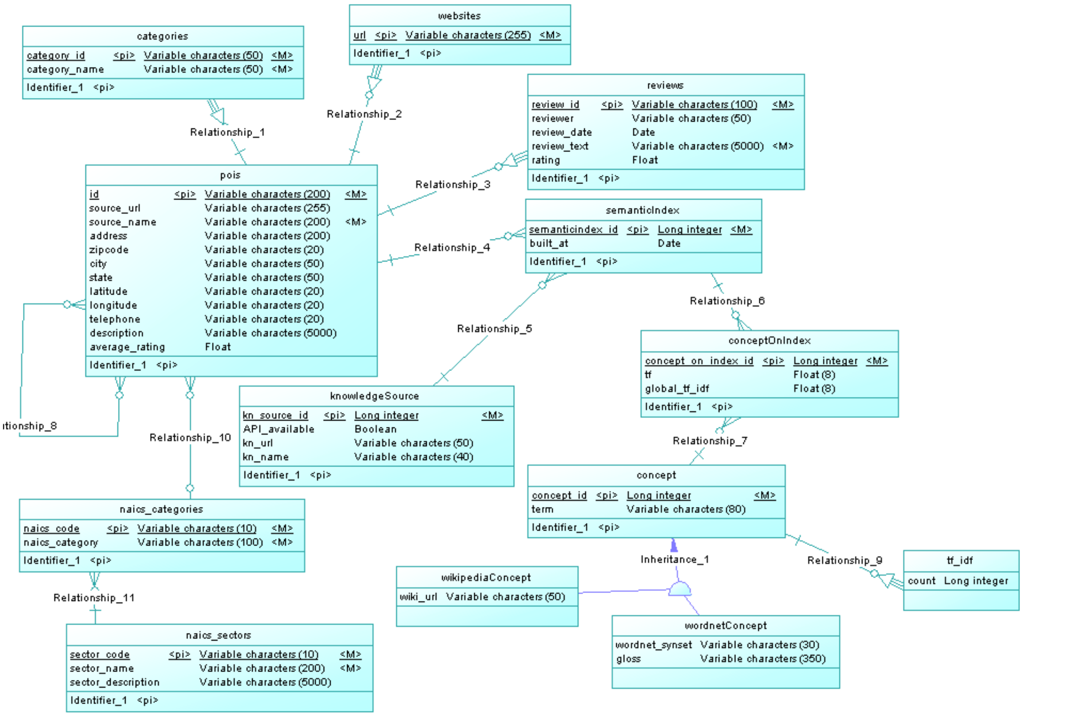

categories
category_id <pi> Variable characters (50) <M>
category_name Variable characters (50) <M>
Identifier_1 <pi>
websites
url <pi> Variable characters (255) <M>
Identifier_1 <pi>
Relationship_1
Relationship_2
reviews
review_id <pi> Variable characters (100) <M>
reviewer Variable characters (50)
review_date Date
review_text Variable characters (5000) <M>
rating Float
Identifier_1 <pi>
pois
id <pi> Variable characters (200) <M>
source_url Variable characters (255)
source_name Variable characters (200) <M>
address Variable characters (200)
zipcode Variable characters (20)
city Variable characters (50)
state Variable characters (50)
latitude Variable characters (20)
longitude Variable characters (20)
telephone Variable characters (20)
description Variable characters (5000)
average_rating Float
Identifier_1 <pi>
Relationship_3
semanticIndex
semanticindex_id <pi> Long integer <M>
built_at Date
Identifier_1 <pi>
Relationship_4
Relationship_5
Relationship_6
conceptOnIndex
concept_on_index_id <pi> Long integer <M>
tf Float (8)
global_tf_idf Float (8)
Identifier_1 <pi>
knowledgeSource
kn_source_id <pi> Long integer <M>
API_available Boolean
kn_url Variable characters (50)
kn_name Variable characters (40)
Identifier_1 <pi>
itionship_8
Relationship_10
Relationship_7
concept
concept_id <pi> Long integer <M>
term Variable characters (80)
Identifier_1 <pi>
naics_categories
naics_code <pi> Variable characters (10) <M>
naics_category Variable characters (100) <M>
Identifier_1 <pi>
Inheritance_1
Relationship_9
tf_idf
count Long integer
wikipediaConcept
wiki_url Variable characters (50)
Relationship_11
naics_sectors
sector_code <pi> Variable characters (10) <M>
sector_name Variable characters (200) <M>
sector_description Variable characters (5000)
Identifier_1 <pi>
wordnetConcept
wordnet_synset Variable characters (30)
gloss Variable characters (350)

Figure 3.1: Entity-Relation Model

Boston). After this stage is complete we update the database with the Top N words that best describe the document/event and the corresponding Term Frequency of each concept.

The last stage is where we compute and update the value of TF-IDF for all the concepts in the database.

3.3 Enrichment

The semantic indexes extracted from the documents using only Kusco as a resource were very poor and provide us little more information about an event than the one provided by the raw description. One way to improve the results was to add another layer to our system. This layer is where the semantic index is enriched by using another source of information to add more semantic knowledge.

So, after the last step mentioned in the last section, the system will try to retrieve the article summaries of the pages available in the Wikipedia[3] for each concept in the semantic index.

Using this approach, all Wikipedia page summaries related to the concepts present in a single event are gathered into one single file and fed into Kusco again, which will result in a new list of concepts ranked by Term Frequency. The last step is to calculate again the TF-IDF for each concept so we can select the best ranked concepts to be used as labels for each event.

3.4 Web Service

As a feature and a better way to integrate all the modules into the team project Web Site, it was developed a Web Service that exposes all the functionalities of the system.

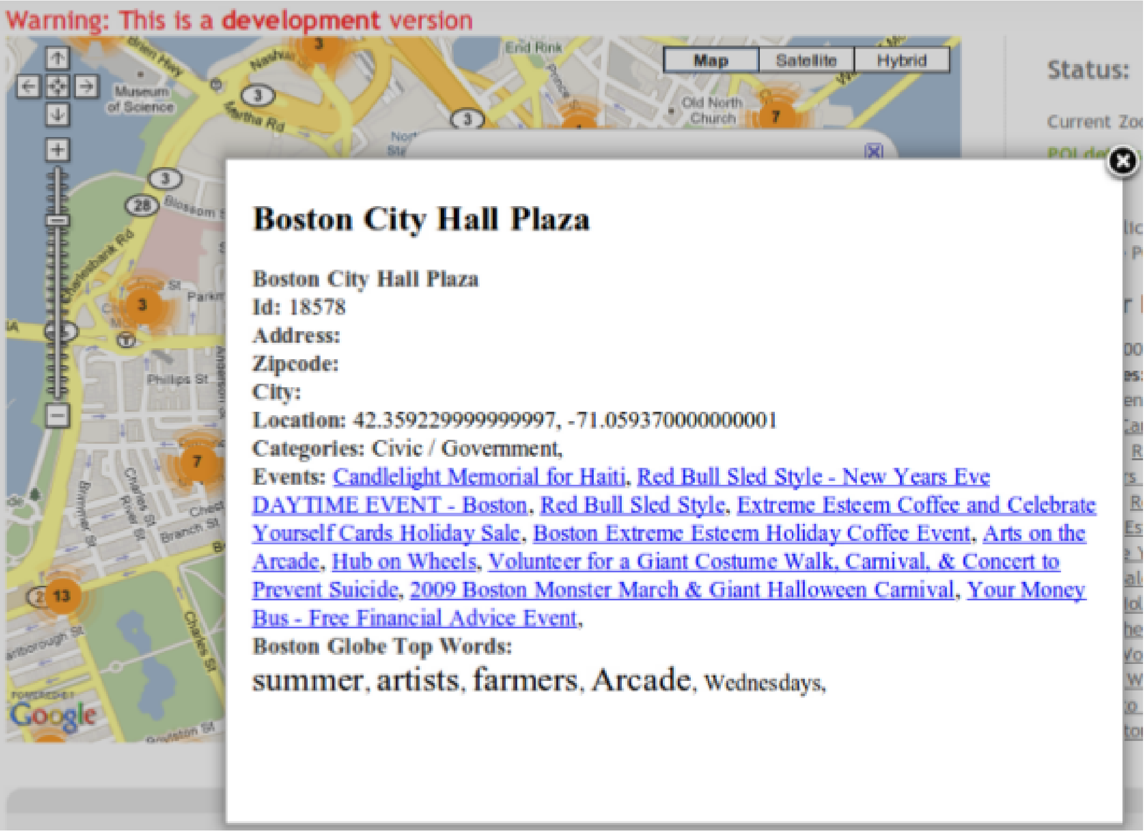
The Web Service follows a RESTfull[4] architecture and provides the user with the ability to get a semantic index that better describes a place or area by specifying a GPS coordinate or a Venue id.

The final semantic index describing a place is achieved by normalizing all the vectors of the documents/events within the venue or area and merging them which results in a vector with concepts ranked by tf-idf.

The main objective of developing this Web Service was to make the process of integrating this service simpler in other future and current services from the research group. Since this web service is also available for the exterior it is also possible for anyone to get semantic indexes extracted from events from Boston city.

[3] http://www.wikipedia.org

[4] http://en.wikipedia.org/wiki/Representational_State_Transfer



Boston City Hall Plaza

Boston City Hall Plaza
Id: 18578
Address:
Zipcode:
City:
Location: 42.359229999999997, -71.059370000000001
Categories: Civic / Government,
Events: Candlelight Memorial for Haiti, Red Bull Sled Style - New Years Eve DAYTIME EVENT - Boston, Red Bull Sled Style, Extreme Esteem Coffee and Celebrate Yourself Cards Holiday Sale, Boston Extreme Esteem Holiday Coffee Event, Arts on the Arcade, Hub on Wheels, Volunteer for a Giant Costume Walk, Carnival, & Concert to Prevent Suicide, 2009 Boston Monster March & Giant Halloween Carnival, Your Money Bus - Free Financial Advice Event,
Boston Globe Top Words:
summer, artists, farmers, Arcade, Wednesdays,

Figure 3.2: Research Group Integration.

Chapter 4

Experiments

4.1 Document labelling

In this chapter we present some examples of the results obtained using our methodology applied to events from Boston. The events here presented are a small subset of our database that currently has 40405 events hosted in 4529 different venues in Boston and were extracted from the Boston Calendar and Zvents sources in a time window starting in August 25th 2009 to January 15th 2010. The average number of events per venue is 9.5 (std. dev. 35.9). There are also some events in our database from the New York city that were extracted from the Yahoo Upcoming service but they are much fewer.

Tables 4.1 and 4.2 introduce some events that are indexed in the database. Table 4.1 contains only events that are considered good examples and table 4.2 contains examples with bad results. First of all, it is important to note that what we distinguish from good and bad events is the verification of the results to check if the semantic index is related to the event description and it provides the user with a richer semantic knowledge about the event. So, in the end, what we consider bad events and venues are those where semantics indexes are presented with a very poor and distant result of semantic knowledge from the original category/topic.

The next table, 4.3, contains the results obtained for each event presented in the previous tables. The column Concepts has the semantic index after the first iteration of Kusco, and the next column has the new semantic index after the enrichment stage is applied.

ID	Name	Description
88168498 A	Nature Trail and Cranberry Bog at Patriot Place	This half-mile trail is part of a 32-acre cranberry wetland system and wooded area with a bridge and observation platform stretching across a six-acre pond. - June Wulff, Globe Staff
87831447 B	Salem Farmers' Market	The Salem Farmers Market is a tradition that dates back to 1634. With it's peak around 1930, The City of Salem is now renewing its tradition of the Salem Farmers Market in downtown Salem, MA. Opening day: June 25, 2009, the 375th anniversary of the birth of the Salem Farmers Market and the rebirth of a Salem tradition. The Salem Market works to Provide a convenient and congenial means of purchasing locally grown or prepared food products and Support local agriculture and producers.
88684905 C	8th Annual Village Cadillac Day	The Cadillac La Salle Club is going to be making their 8th annual appearance at Ray Ciccolos Cadillac Village of Norwood. The club is expected to bring over seventy antique Cadillacs from the 1920's up through the 70's. This is event is FREE and open to the public, and will feature a DJ and free refreshments
87036119 D	Treasures from The Boston Athenaeum	The paintings, sculptures, drawings, photographs, and manuscripts in this summer installation draw from the collections of the Boston Athenium and add to the wealth of objects always on public view on the Athenium first floor. Over 40 artists are represented, ranging from Italian and Scottish to American and from the 16th to the 21st century. The objects on view are as varied in style as in subject matter, and include: a portrait by the 16th-century

Table 4.1: Good examples of Boston events.

ID	Name	Description
88210593 E	EPOCH of Chestnut Hill campus sponsors month-long food drive	EPOCH Senior Healthcare of Chestnut Hill and EPOCH Assisted Living at Boylston Place, will be collecting non-perishable food items in their lobbies throughout the month of August. EPOCH will donate the food collected to the Brookline Food Pantry. Contact Mary Rivera at 617-243-9990 for more information.
88513856 F	Fall Forest Festival	10:00am - Noon: Volunteer 1:00 - 4:00pm: Games, tree climbing, and family nature walks in the woods 12:00 - 2:00 pm: Landscape watercolor painting workshop - all materials provided / on Schoolmaster Hill ***Meet at the Resting Place / Shattuck Picnic Grove Bus Route #16 from Forest Hills (Behind Shattuck Hospital across from Forest Hills Cemetery)
87899964 H	Emerald Society of Boston Police Dept.Halfway to Saint Patricks 5K Road Race Run	Come early and see Army Blackhawk Helicopter, Humvees, Playstation on Jumbotron and more. A benefit by cops for kids with cancer.
87090680 I	Lexington Farmers' Market	Lexington Farmers Market, corner of Massachusetts Ave, Woburn St., and Fletcher Ave. in Lexington Center. Tuesdays, June 9 through October 27, 2009, 2-6:30 p.m., rain or shine. Features locally grown produce, a variety of meats, fish, baked goods and other prepared foods, and artisans tent. Admission free. For more information, and to subscribe to the weekly newsletter, visit www.lexingtonfarmersmarket.org

Table 4.2: Bad examples of Boston events.

ID	Concepts	Wiki Concepts
A	Globe, system, plataform, trail, pond	Pond, Falls, streams, currents, winter
B	Farmers, birth, rebirth, anniversary, agriculture	cultures,consumption, carbohydrate, Food safety, gastronomy
C	Cadillac, Norwood, apperance, refreshments, Cadillac La Salle Club	Cadillac, Michigan, Automobile, General Motors Company, vehicles
D	objects, Boston Atheneum, portrait, sculptures, collections	Paintings, Eastern, scenes, Sistine Chapel, Mona Lisa
E	Boylston, Chestnut, Healthcare, items, lobbies	Monoclonal, Surgery, Medicine,Dentistry, health systems
F	Forest Hils Cemetery, Volunteer, Games, Noon, materials	Trees, Collins, plants, Macmillan, Sequoia sempervirens
I	Jumbotron, kids, Playstation, benefit, cancer	Cancer, cells, abnormalities, neoplasm, treatment
H	meats, Massachusetts, Fletcher, tent, Lexington	tent, camping, shelter, rope, poles

Table 4.3: Boston Events Top Five Concepts.

By looking at the tables it is possible to recognize that the system was capable of extracting new concepts that were not available in the original description. This happens because we added semantic knowledge by processing the Wikipedia summaries for each concept of the first semantic index.

On the other hand, the last four events have very poor results which are justified by the lack of information related to the event in their descriptions. It is also possible to notice that some words are noise introduced by the user description or the case of a faulty screen scraping. In other times, this happens because the description is so small that the term frequency of the terms are nearly equal, thus the ranking is not efficient.

The last table 4.4 introduces the semantic indexes of four places that were computed by merging the semantic indexes of events held in the same place. As it is possible to notice, the more documents/events used to calculate the result, the better the definition will be. The first place in the table is a Point of interest extracted from Boston Calendar that is used to represent the city and some events without a specific hosting place.

[1] Hebrew Bible

[2] Massachusetts Institute of Technology

Name	**Concepts**	**Num Docs**
City of Boston	traffic, boston, competition, intersection, lanes, vehicle, rivals, frredom trail	11
Tremont Temple Baptist Church	bible, category judaism, tanahk[1], prayer, language, christians, meditation	21
New England Aquarium	aquaria, presentation, animals, fur seals, turtle	135
MIT[2]	community, dance, massachusetts, questions, students, seminar, lecture, university, skills	270

Table 4.4: Boston Places.

4.2 Pattern Exploration

In this section it is explained an experimentation that was done with the semantic indexes with the objective to extract simple and complex patterns from the flow of events in a specific place and time. To achieve this, the best idea was to implement a clustering algorithm and merge the documents that are semantically close to the same cluster and as a final result we would get a list of clusters where each cluster is probably related to a topic/category of documents/events.

The most used algorithm, as argued by Zhao et al[34], for clustering documents by their similarity is the Hierarchical Clustering because it is capable of building meaningful hierarchies out of a large collection of documents which are ideal for providing data-views that are consistent, predictable and at different levels of granularity. While other algorithms like K-means require that we provide them with a variable that represents the number of clusters that we want to get as a result, the hierarchical algorithm does not require that because it has a different approach for extracting the clusters depending on a value that represents the limit and granularity of the clusters. And since we do not know how many clusters/patterns would result from the input, this is the best approach.

After the clustering algorithm was implemented, it was required to implement a set of document similarity measures that would act as the distance function for documents and centroids of the clusters. The similarity measures implemented for this experiment was the ones described in section 2.3.

The methodology for this experimentation consisted in two approaches. The first approach was to find similar events inside a venue. To achieve this we have gathered

	Average	**Standard Deviation**
Events after Enrichment	52	27
Events before Enrichment	47	30
Venues after Enrichment	45	35
Venues before Enrichment	41	38

Table 4.5: Events and Venues Clusters and Patterns.

	Average	**Standard Deviation**
Events after Enrichment	31	23
Events before Enrichment	24	25
Venues after Enrichment	33	33
Venues before Enrichment	28	35

Table 4.6: Clusters and Categories.

a list of 200 venues from our database and for each one gather the events that were held there in a time window from September 2009 to the June 2010. After we gather all this information, we extracted the semantic indexes for each event and computed the *tf-idf* value for each concept inside the index but used as a corpus only the events of that venue. This last measure would affect the final *tf-idf* of each concept because it affects directly the *idf* variable of the function. The reason we take this choice is because it does not make sense for the semantic index of an event to take into account the concepts that are used outside of the venue if the main objective is to extract knowledge from the venue only.

The second approach was very similar, but instead of trying to find similar events inside a specific venue, the objective was to find patterns of events inside a radius of meters. So, what we have done was running the algorithm for each location of the venues and instead of restricting the events to the venue we restricted it to a radius of 500 meters.

Because we wanted to test and compare the different document similarity functions, we executed the experimentation several times for each venue, which would consist in 4 times for each function multiplied by 10 steps. Each step represents the maximum limit of similarity of documents we want for a document/event to merge with a centroid.

Table 4.5 presents us with the results of average and standard deviation of clusters that were computed from the events and venues approaches, before and after the enrichment process is applied. And in the table 4.6 it is presented the average and standard deviation of the difference of clusters obtained and the number of distinct categories of the events used.

The conclusions that we can get from these tables is that, as we expected, the taxonomy of categories of events and venues used to classify each event is very poor because our experiment just proved that there are significant differences between documents for them not to be clustered with each other. Another aspect is that this distance from the number of clusters and categories could have been greater if we would not limit the number of top concepts in the semantic index to five. It is not possible to make other conclusions or assumptions about these results because there are multiple variables. One of these variables is the threshold used as a limit for the document similarity. We could not say what is the best using this approach. To get the answer to this we would probably need to use volunteers to validate the results. Another question to answer is find which distance function (document similarity function) is better. From the results we have obtained we can see that the functions that get the minimum number of clusters are the ones that take into account the weight of each concept. This makes sense as it was explained in section 2.2.

It is also important to see that main objective of this experimentation is to prove that since we can see which events are similar we can apply other layers of clustering to extract more complex patterns. For example, it would be possible to extract that each first Saturday of each month there would be held a music concert at a specif restaurant if we would apply another layer of the clustering algorithm but using a distance function that would compute the time difference between events. The reason why we did not explore this was because we do not have enough data to compute this kind of patterns since we would need to find the frequency of the *episode* of events, to consider it as a pattern. Instead we have executed this experimentation to serve as a proof of concept for this kind of application.

This kind of pattern information can be very useful for some business models that directly depend on others. For instance, a company can move their resources in a better and predicted way with the objective to advertise or sell their products at a place if they know that an event will be held there in a near future that will gather people from their target public.

Chapter 5

Validation

5.1 Category stability

5.1.1 First approach

This research project faces an important challenge of understanding the actual quality of the results in terms of the correctness of the words assigned to places. The list of words that best describes a place is by nature subjective, because as referred above in chapter 1, a place can be defined according to different perspectives, and each perspective can vary with subject. In terms of validation, this raises difficult questions even for the typical user survey. The only way to guarantee a good validation using human resources is using a large sample of people, making sure that they know all the places, which then becomes unpractical.

Thus, we decided to analyze our results according to category consistency. Each POI has one or more category, so the task is to verify the stability of the word patterns according to those categories. The first approach is to apply a clustering algorithm such as K-Means, where K corresponds to the number of different categories. After clustering with a training set, it is applied a classification task that consists in categorizing with one of the clusters using the semantic index of the event.

As we can see in table 5.1 the results are somewhat poor. This implies that either the word patterns are not stable with respect to category or they are more elaborate than achievable with clustering algorithms.

Algorithms	**Percentage**
Rand. baseline	17.3%
Fixed baseline	13.86%
K Means	24.93%
Bayes Network	51.08%

Table 5.1: Statistical Results.

However, Bayes Networks, which are actually more common in text categorization, presented the best results (accuracy of 51.08%). This analysis just proves that difficulties exist that are inherent to systems dealing with unstructured text. In this case, two events can match two different categories that are related to the same subject. For instance, one event talks about Italian food and another event talks about food in general, though they relate to the same topic. One way to cut this problem is to apply *Concept Similarity* between two categories and if we get a high value we can assume that they are nearly the same, thus resulting in a higher positive match results.

5.1.2 Second approach

In this section we explain the second approach that was taken in order to try to make a validation model that was in part automatic and at the same time that was capable of doing a better validation than the first approach explained in the previous section. To achieve this we needed to develop a algorithm that would take into account the semantic indexes and the weight of each concept in order to be able to do reasonable evaluation of similar documents and classify with a closer category or topic.

One of the best algorithms for this type of classification is the K Nearest Neighbor if it is well adapted with a weight system. Actually, Eui-Hong Han et al argued that the study they made proved that a well adapted Weight Adjusted Nearest Neighbor Classification algorithm can outperform other algorithms, such as C4.5, RIPPER, Naive-Bayesian, PEBLS and VSM[35]. The reason for these results is because this algorithm finds the k documents that are closer to the document to classify and those k categories of the documents “vote” for the category of the new document.

The second step of this approach and the main problem of the previous approach was to discover a way to find if the classified category and the real category are closer enough semantically to considering them the same, and consequently a positive match. To achieve this we have also developed a set of algorithms that measure the similarity between concepts and that we described in the section 2.4. Although we have developed all these measures we only have used six of them: *google distance, path similarity, Wu-Palmer, Jian-Conrath, Lin and Edit Distance.* The main reason why we do not use the other two is because *lch* was very slow and therefore a major bottleneck for the whole validation system, and the *resnick* was not used because the *jcn* and *lin* measures are already based in the *resnick* formula. For measuring the distance between the documents inside the *kNN* algorithm, we have used the document similarity functions described in section 2.3.

We ran the algorithm 240 times for each document, which in this case it could be an event or a venue, because we also alternated the variable k of the number of neighbours from 1 to 10. The dataset that we have used for this workload was the same that was used for the volunteer validation, discussed in the next section (5.2).

Since we use many variables, we can not make a real estimation of the positive and negatives match in the classification process without defining a fixed value to the majority. And if we do this we do not have guaranty of correctness because we can not assume that, for example, two categories are similar if the *google distance* is lower than 0.5. This happens for three reasons: the first one is because some of the term similarity functions depend on *Wordnet*, and so, if the category *lemma* is not present then there is not a distance result; the second one is because even if we used the other functions we do not know what is the best threshold value to use as a limit to positive and negative match. If we have done this, then we needed to validate those results with the help of volunteers. Third, and last, we also do not know what is the best number of neighbours to be used in the algorithm.

Taking these aspects into account, we present here some of most challenging results and analyse them.

In the table 5.2 it is possible to see some of the examples of results that we have classified and that at a first sight they sight like negative matches.

For instance, the example with id *730VB*, which is a semantic index before the enrichment process was applied to the venue, has a real category of Theater but the *kNN* algorithm classify it as a College / University using the cosine distance as well as for all values of k neighbours. At first it seems a wrong classification, but if we make further analysis we have discovered that the venue was badly categorized in

[1] Category obtained after running classified via kNN.

[2] Real category extracted from the source

[3] Document similarity function used

ID	**Cat**[1]	**Cat**[2]	**Func**[3]	**k**
730VB	College / University	Theater	cosine	1
766VB	School	College/University	cosine	4
768VB	Nightclub	Club	consine	1
813VB	College/University	College University	cosine	1
813VB	Non-profit	College University	consine	7
820VA	Library	BookStore	cosine	1
817VA	Ballroom/Dance Hall	Community Center	cosine	1
965VA	Theater	Arts/Cultural center	cosine	1
809EA	music	rap/hip-hop	cosine	1
1225EA	jazz	jazz	cosine	1
1225EA	visual arts	jazz	jaccard	1

Table 5.2: kNN Category classification results.

ID	**google**	**path**	**wup**	**jcn**	**lin**
730VB	0.56	0.07	0.14	0.05	0
766VB	0.27	0.14	0.57	0.10	0.49
768VB	0.32	0.06	0.11	0.05	0
813VB	0.45	0.08	0.25	0.05	0
820VA	0.48	0.11	0.6	0	0
817VA	0.51	0.07	0.13	0.05	0
965VA	0.28	0.67	0.13	0.07	0
809EA	0.35	0.11	0.42	0.09	0.37

Table 5.3: kNN Category similarity.

the source because the venue name is in fact “Brandeis University”. So, the system was able to correctly classify the venue even with the wrong category applied at the resource source.

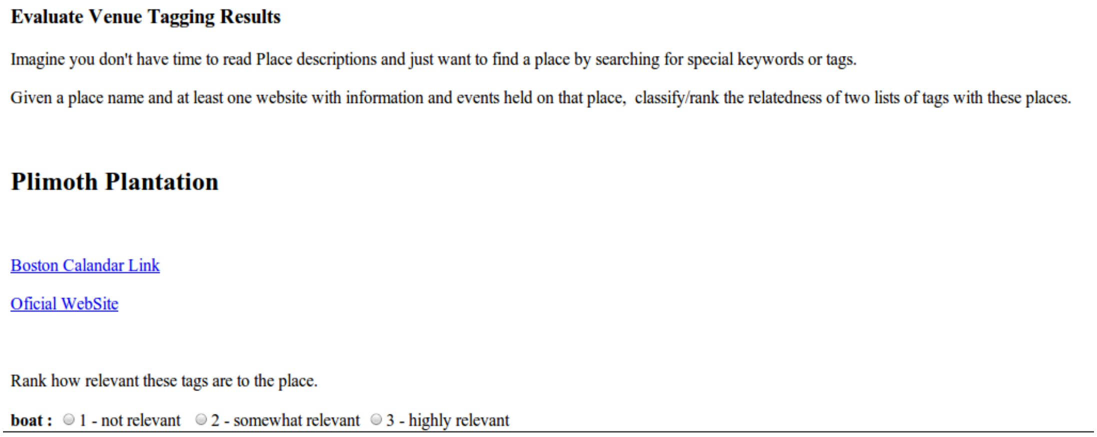
Other good examples are the ones with the id *813VB*. We only used one neighbour to classify the venue, the result was a positive match, but when we used 7 neighbours to vote the right category we get another category. This is normal, and in fact it does not mean it is wrong, because a college is really a non-profit institution. It is also possible to prove this if we analyse the table 5.3 where it is possible to view all the distances between the two categories from the distinct similarity functions.

All the values returned from the functions in this table fall inside the interval of [0..1], where 0 means that the two categories are not similar and 1 that they are similar, except for the google distance, where 0 means similar categories. This table shows, at a first analysis, that the similarity functions that use *Wordnet* and the Information Content (IC) do not offer the same performance as the as the *google distance*. The main reason for this is because the *google distance* does not depend on a lexical resource as Wordnet and the relationships between synsets.

In conclusion, we can say that this methodology of classification has proven to be correct by the use of this examples. But on the other hand, it can be noticeable some problems when the number of *neighbours* allowed to vote is too small or too large. Another conclusion we can take from the results obtained and posterior self validated is that the concept similarities functions *cosine* and *tanimoto* have outperformed the other two. This happened because the jaccard and overlap formulas do not use the weight of the concepts in the semantic index, and therefore are measures of binary scheme. This means, for example, that two events that share the same two topics but each one focus on a different topic would be classified as similar even with the major differences. This type of behaviour does not happen when the *consine* or *tanimoto* functions are used. Another sight that could be made from the results is that for most part of the test cases, the events and venues would be classified with the same category with all values of k neighbours. This is normal because we only use the top five concepts ranked by *tf-idf* for each semantic index, and thus, the probability of making a minor similarity measure with another semantic indexes that are poorly related is very small.

5.2 Mechanical Turk

Because we are working with Natural Language Processing systems, using good validation models is difficult and it is nearly impossible in order to make certain conclusions or assumptions. Taking this into account, the only method to do a good validation is by making use of human intelligence trough humans. This is why we decided to use the Amazon Web Service, Mechanical Turk, which makes possible to publish our data in their servers and having multiple persons from multiple places in the world to validate our system in turn of a small cost[36]. The important aspect to be careful is that there exists some users that are spammers and Amazon does a good work by providing the correct tools so we can choose which type of people can participate in our validation. Some of these properties, related to the user, that we can choose are the positive feedback and the country. Other way to remove spam data from the validation is to make the same question to different users. The variable that corresponds to this property is called *quorum*. Because we wanted to make sure our results were free of spammers we only let users with a 95% positive



Evaluate Venue Tagging Results

Imagine you don't have time to read Place descriptions and just want to find a place by searching for special keywords or tags.

Given a place name and at least one website with information and events held on that place, classify/rank the relatedness of two lists of tags with these places.

Plimoth Plantation

Boston Calandar Link

Oficial WebSite

Rank how relevant these tags are to the place.

boat : ○ 1 - not relevant ○ 2 - somewhat relevant ○ 3 - highly relevant

Figure 5.1: MTurk Hit preview.

Options/Enrichment	Before	After
Most Relevant	43%	8%
Relevant	46%	40.9%
Nothing Relevant	10%	50.5%

Table 5.4: Statistical Results - Events Batch 1.

feedback participate and we choose to make the same questions to 3 distinct users, so in the final analysis we can select the best of 3 responses for each event. In this section we only present the results after we apllied the filter of *best of 3*, but it is also possible to view and analyse the charts withoud the filter applied in the appendix B.

We ran two types of batches in MTurk, and each type of batch consisted of two batches (one for events and another for venues), making a total of 4 batches executed in Mturk. In the first 2 batches we provided the user with the following data: an event/place name, an event/place description, official website if provided and 2 lists of concepts (semantic indexes), one before wikipedia enrichment and another after. It is possible to see the type of screen that the users used to classify the events and venues in figure 5.1.

Each semantic index is composed by the top five concepts ranked by TF-IDF. With this, the user was asked to classify the relevance of each semantic index within 3 levels: Nothing Relevant, Relevant, Most Relevant. In both batches we provided 960 events and 200 venues to be validated by 69 and 19 distinct users, respectively.

Table 5.4 and figure 5.2 shows the results that we obtained from Mechanical

Options/Enrichment	Before	After
Most Relevant	38%	33%
Relevant	46%	53%
Nothing Relevant	11.5%	14%

Table 5.5: Statistical Results - Venues Batch 1.

	Events	Venues
Improved	10.7%	26.5%
Worsen	64.6%	29.5%
Mantain	24.5%	44.5%

Table 5.6: Statistical Results - Batch 1.

Turk.

As we can see, the results obtained for the events before the enrichment with wikipedia are relatively good considering that only 10% are classified as nothing relevant with the event. But after the enrichment the values dropped significantly and this can be explained because while we are trying to improve the semantic information in the list of concepts we are also introducing possibly noise, and this can be analysed in table 5.6 and figure 5.5 where we can see that when we applied the enrichment process, 10.7% of those semantic indexes have improved, 64.6% worsen and 24.5% maintained the relevance score. These results can be explained by the fact that the events were ranked by TF-IDF using as corpus the whole database of events, and this corroborates with the results obtained for the venues because with the venues the corpus is a subset of events (events held in the venue). The results for the venues can also be viewed in table 5.6 and figure 5.4. In this case we can see that the results are almost equal when applied the enrichment process, but if we analyse the chart in figure 5.4 we can see that 25% of the venues decreased their relevance rank by 1 level and others 25% increased 1 level. Other aspect is that nearly half of the venues stayed in the same relevance rank after the enrichment process was applied. The conclusion we can get from these results is that the enrichment process works as it was expected, but if the semantic index contains a concept that is not relevant for the event or even noise, then the process ends up adding noisy information.

We notice in the charts that present the improvement detected from the enrichment process, that the labels of the bars that go from the values -2 through +2 consist in the improvement detected. So, if a bar with label -2 has a value of 25% then we can say that 25% of the events have decreased rank classification by to levels (from Most Relevant to Nothing Relevant).

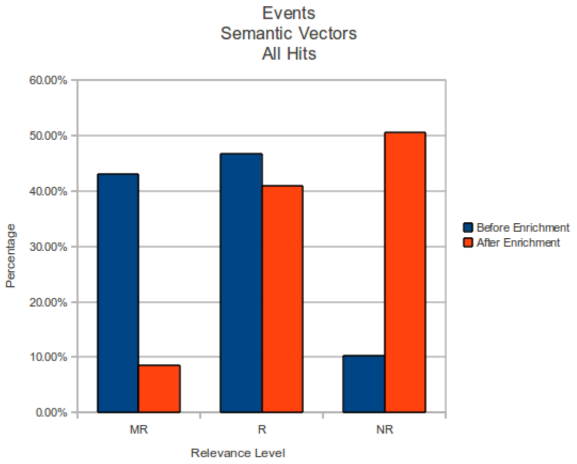


Figure 5.2: MTurk - Events Batch 1.

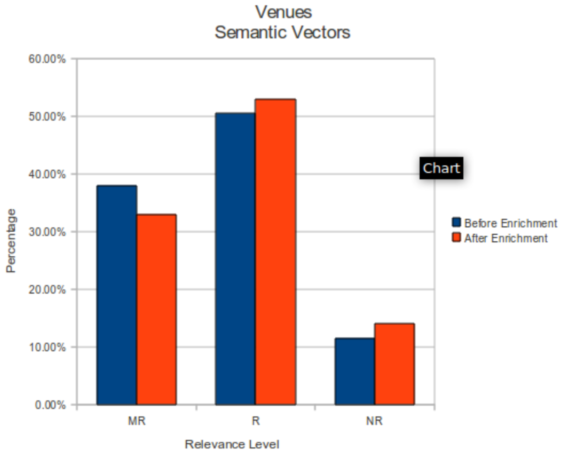


Figure 5.3: MTurk - venues Batch 1.

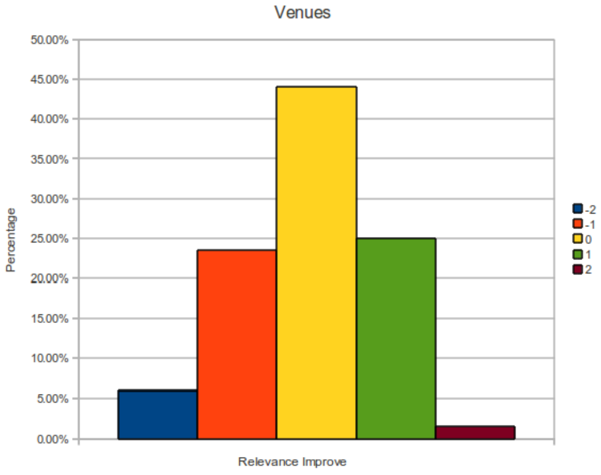


Figure 5.4: MTurk - venues Batch 1 improve chart.

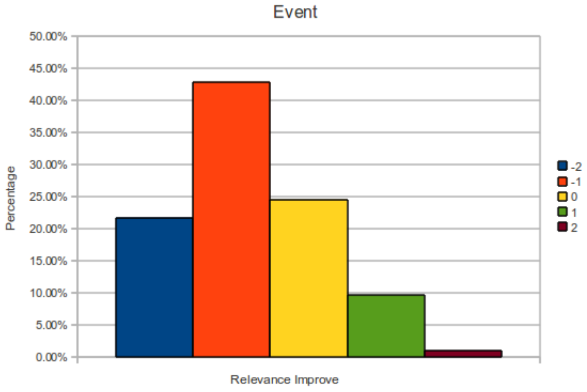


Figure 5.5: MTurk - Events Batch 1 improve chart.

If we limited the corpus of documents used in the events batch by a geographic radius instead of using all the events in our database, then probably we would have better results. On the other hand, this would introduce some problems in the performance of the overall system and require a careful study and implementation.

The second batch that we ran in MTurk was with the same data, but instead of asking the user to classify the relevance of each semantic index, we asked them to classify the relevance of each concept in the semantic index. The reason we chose this approach was because it was very difficult to interpret the results in cases where the semantic index as voted by three different users with 3 different choices of relevance. We do not know for sure if this happened because one or two concepts in the semantic index was noisy and would affect the choice of the user in different

Relevance	**MR**	**R**	**NR**
Concept 1 BE	66.5%	28.6%	4.7%
Concept 2 BE	65.7%	26.3%	7.9%
Concept 3 BE	61.2%	26.4%	12.2%
Concept 1 AE	35.9%	26.1%	37.9%
Concept 2 AE	34.1%	24.1%	41.6%
Concept 3 AE	22.5%	33.8%	43.6%

Table 5.7: Statistical Results - Events Batch 2.

Relevance	**MR**	**R**	**NR**
Concept 1 BE	36%	58.5%	5.5%
Concept 2 BE	59%	37%	4%
Concept 3 BE	66%	30%	4%
Concept 1 AE	67.5%	28%	4.5%
Concept 2 AE	61.5%	32%	6.5%
Concept 3 AE	36%	54.5%	9.5%

Table 5.8: Statistical Results - Venues Batch 2.

ways. Another change we made in this batch was the number of concepts in the semantic index that decreased from 5 to 3 mainly because of the cost associated to the MTurk service. The number of users who validated this batch was 103 for the events batch and 24 for the venues batch.

In the tables 5.7 and 5.8, we can see the results obtained for the second batch for events and venues respectively. It is also possible to analyse this information from the figures 5.6 and 5.7. In the appendix A there are other charts that have the remaining results about he improvement level obtained for each concept for this second batch. Each table is composed by a first row that contains the levels of relevance: Most Relevant (MR), Relevant (R), Nothing Relevant (NR); and the columns refer to the first 3 concepts of the semantic index before enrichment (BE), followed by the list of three concepts after enrichment (AE).

These results tell us what we suspected before. The users were having doubts on how to classify the semantic index in the first batch if the event/venue was bad. This means a semantic index with poor semantic information or a concept that was only noise (eg. HTML tags).

We can see that on the events and venues batch only a small subset of concepts were classified as not relevant. But we also have to take into account that we dropped the last two concepts from the semantic index and that may have some influence in

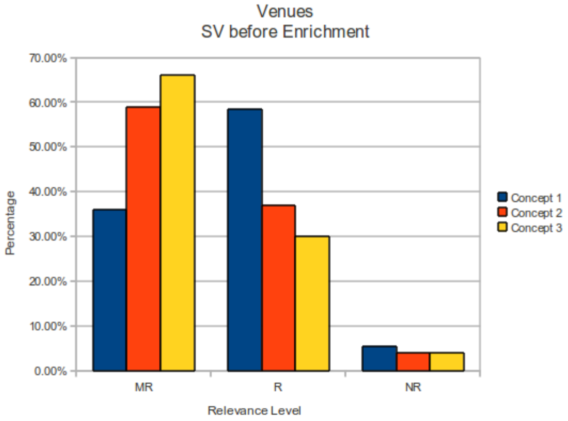


Figure 5.6: MTurk - venues Batch 2 - Before Enrichment.

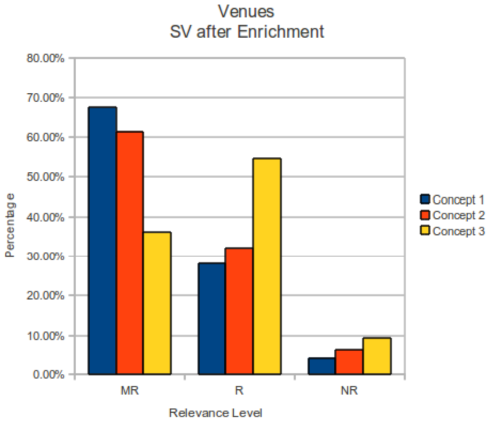


Figure 5.7: MTurk - venues Batch 2 - After Enrichment.

these results. Another important aspect is that on the venues batch, the process of enrichment has very good results: the concept that take the first position makes an improvement from almost 30% to 70%, but this tend to decrease for the following concepts which makes sense because of the TF-IDF ranking system. On the other hand, the events batch presents us with some low results, mainly because of the same reason as explained in the first batch, which is the use of all events for the corpus to calculate the *IDF* value.

Finally, from chart 5.6 it is possible two sight that before the enrichment process is applied, the order by which the concepts had the *most relevant* status is in reverse to what it would be expected. After the enrichment, chart 5.7 shows that the results are now it the order which makes more sense. Again, this is explained by the use of the vector space model. Because we are talking about venues, we only use as the corpus of documents , the events that are held in that venue, therefore, if the venue only has 10 events then we only have 50 concepts in total (1 semantic index has 5 top concepts) and this affects the *tf-idf* (see section 2.2) value. After the enrichment process is applied, the number of concepts per event is increased drastically, and consequently the *tf-idf*.

Chapter 6

Conclusions

In this report, we presented a methodology for the extraction of semantics of places and events from online resources with the intent of understanding urban dynamics of the city.

Recalling the objectives described in section 1.2, it is possible to sight that each one of the objectives was addressed. We have added two new sources, in addition to the initial Yahoo Upcoming source, and improved the information retrieval code in order to reduce the noise coming along with the useful information. In addition to the methodology used to extract semantic indexes from events, we have also studied and developed a way to merge these semantic vectors and produce new ones capable of identifying places, venues or even areas in a unique way. As a final functionality of the system we have also added another perspective working as a layer for improving the original semantics indexes by enriching the semantic information. We achieve this by using Wikipedia.

We have also validated the methodology using two different approaches: one with volunteers around the world where we analysed the relevance level of each concept in the semantic index with the event or venue; and a second validation model capable of analysing stability between the concepts and event/venue categories.

In the last task of the work plan we analysed the flow of events in order to study and implement an algorithm capable of finding simple and complex patterns. We have concluded that the data retrieved from our sources since the start of the internship do not provide us with enough pattern frequency so that we can reach to valid conclusions, therefore, the algorithm only acts as a proof of concept.

Although the methodology presented in this dissertation works with information provided in the English language, it is important to take into account that it is possible to adapt the system to other languages. The only required change to make is to adapt the Concept Extraction application to the target language. In our system this application was Kusko (section 2.5.1).

In practical terms, what we have developed is a methodology capable of retrieving semantic indexes that better describes a place. These tags or semantic index can be useful for various applications, namely, POI search, context-aware applications on ubiquitous systems, automatic advertising.

Results also show that despite the problems and difficulties inherent to systems using Natural Language Processing for unstructured texts, it is possible to obtain meaningful descriptions of place from dynamic web sources.

6.1 Future Work

There are some ideas that can be explored in order to improve our methodology and system. Some are related to the performance of the system which is dealing with so much information that is important to scale the platform.

Other ideas are related to a better way to integrate perspectives/sources. It may be possible to set a weight to each perspective so we can define which perspective is more important. After this feature is implemented we can try to improve it to adapt the weights dynamically.

Other aspect to explore explore is the semantic information that can be retrieved from the links of Wikipedia. With this information we can know how each concept relates to each other and possibly find new patterns between documents/events, or even improve the semantic index by inferring new concepts. The main reason why this was not implemented is because it makes it necessary to parse the whole database of wikipedia which is nearly impossible to do in a reasonable time frame without a cluster. This research idea could lead to another sub projects where we can even build an ontology similar to *Wornet* but in an automatic way.

Finally, another idea is to classify a place based on its events dimensionality. That is, trying to infer other concepts from bursts of regular events. For instance, a stadium that hosts different types of sports depending on the season.

Bibliography

[1] S. Hacker, "White paper: Trainable semantic vectors & semantic signatures," tech. rep., TextWise, 2008.

[2] Calais, "Opencalais whitepaper," tech. rep., Thomson Reuters, 2008.

[3] D. Deng and R. Lemmens, "Web 2.0 and semantic web: Clarifying the meaning of spatial features," 2008.

[4] T. Rattenbury, N. Good, and M. Naaman, "Towards automatic extraction of event and place semantics from flickr tags," in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 103–110, ACM, 2007.

[5] B. Sigurbjörnsson and R. van Zwol, "Flickr tag recommendation based on collective knowledge," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, (New York, NY, USA), pp. 327–336, ACM, 2008.

[6] E. Margolis and S. L. A. O. or Mental Representations?, "The ontology of concepts."

[7] F. C. Pereira, A. Alves, J. Oliveirinha, and A. Biderman, "Perspectives on semantics of the place from online resources," *International Conference on Semantic Computing*, vol. 0, pp. 215–220, 2009.

[8] J. Hightower and G. Borriello, "From position to place," 2003.

[9] S. Asadi, X. Zhou, H. Jamali, and H. Mofrad, "Location-based search engines tasks and capabilities: A comparative study," 2007.

[10] D. Ahlers and S. Boll, "Location-based web search," in *The Geospatial Web*, Advanced Information and Knowledge Processing, pp. 55–66, Springer London, 2007.

[11] A. O. Alves, F. C. Pereira, A. Biderman, and C. Ratti, "Place enrichment by mining the web," in *Ambient Intelligence*, vol. 5859 of *Lecture Notes in Computer Science*, pp. 66–77, Springer Berlin / Heidelberg, 2009.

[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

[13] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly, 2009.

[14] "Natural language toolkit." `http://www.nltk.org/`.

[15] "Crunchbase database." `http://www.crunchbase.com/`.

[16] E. Brill, "Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging," *Comput. Linguist.*, vol. 21, no. 4, pp. 543–565, 1995.

[17] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," tech. rep., Ithaca, NY, USA, 1987.

[18] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *CoRR*, vol. cmp-lg/9709008, 1997.

[19] A. Budanitsky, "Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures," 2001.

[20] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity: measuring the relatedness of concepts," in *HLT-NAACL '04: Demonstration Papers at HLT-NAACL 2004 on XX*, (Morristown, NJ, USA), pp. 38–41, Association for Computational Linguistics, 2004.

[21] R. L. Cilibrasi and P. M. Vitanyi, "The google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 370–383, 2007.

[22] B. D. Martino, "An ontology matching approach to semantic web services discovery," in *Frontiers of High Performance Computing and Networking ISPA 2006 Workshops*, vol. 4331 of *Lecture Notes in Computer Science*, pp. 550–558, Springer Berlin / Heidelberg, 2006.

[23] B. Bagheri, H. Abolhassani, and H. Sayyadi, "A neural-networks-based approach for ontology alignment,"

[24] A. Alves, F. C. Pereira, A. Biderman, and C. Ratti, "Place enrichment by mining the web," in *Proc. of the Third European Conference on Ambient Intelligence*, 2009.

[25] G. A. Miller, "Wordnet: A lexical database for english," *Communications Of The ACM*, vol. 38, pp. 39–41, 1995.

[26] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," pp. 73–78, 2003.

[27] TextWise, "Semantic hacker api." `http://www.semantichacker.com/`.

[28] "Open directory project." `http://www.dmoz.org`.

[29] Reuters, "Opencalais api." `http://www.opencalais.com/`.

[30] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins, "Visualizing tags over time," in *WWW '06: Proceedings of the 15th international conference on World Wide Web*, (New York, NY, USA), pp. 193–202, ACM, 2006.

[31] A. Jaffe, M. Naaman, T. Tassa, and M. Davis, "Generating summaries and visualization for large collections of geo-referenced photographs," in *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, (New York, NY, USA), pp. 89–98, ACM, 2006.

[32] E. Amitay, N. Har'El, R. Sivan, and A. Soffer, "Web-a-where: geotagging web content," in *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 273–280, ACM, 2004.

[33] P. Serdyukov, V. Murdock, and R. van Zwol, "Placing flickr photos on a map," in *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 484–491, ACM, 2009.

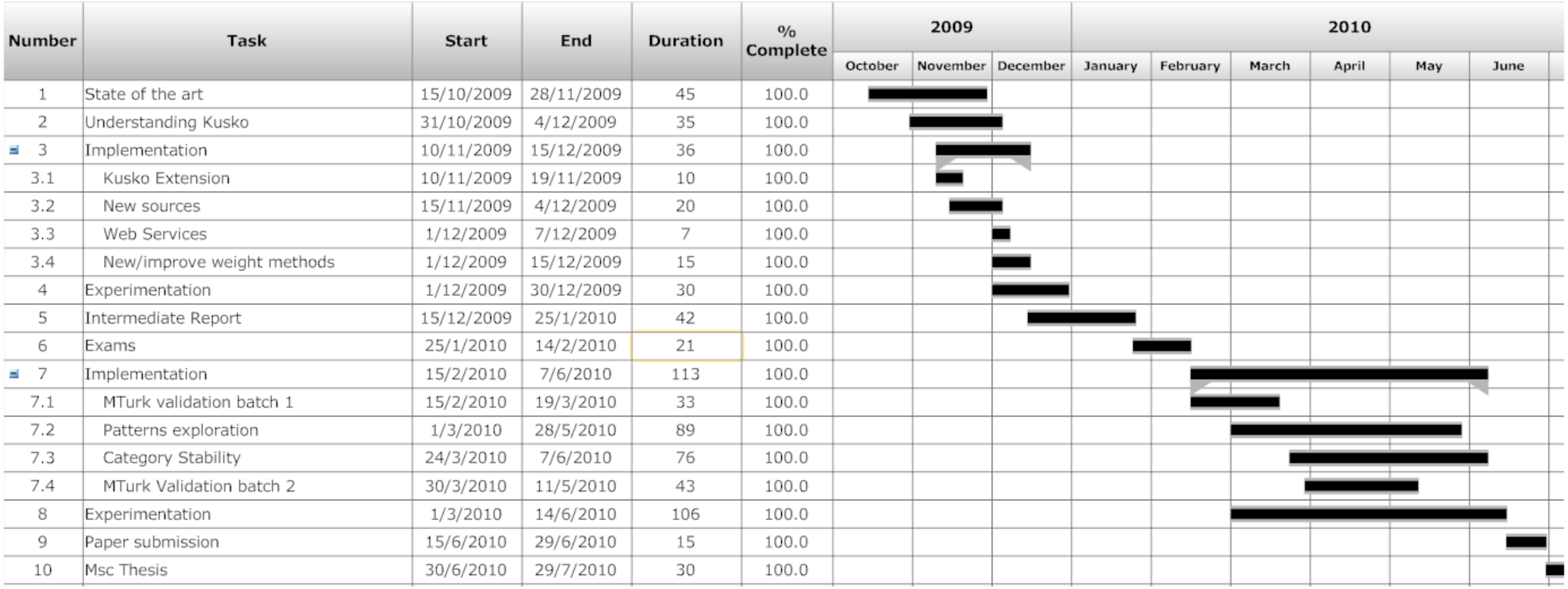
[34] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery*, vol. 10, pp. 141–168, March 2005.

[35] E.-H. S. Han, G. Karypis, and V. Kumar, "Text categorization using weight adjustedk-nearest neighbor classification," in *Advances in Knowledge Discovery and Data Mining*, vol. 2035 of *Lecture Notes in Computer Science*, pp. 53–65, Springer Berlin / Heidelberg, 2001.

[36] O. Alonso, D. E. Rose, and B. Stewart, "Crowdsourcing for relevance evaluation,"

Appendix A

Planning



Number	Task	Start	End	Duration	% Complete
1	State of the art	15/10/2009	28/11/2009	45	100.0
2	Understanding Kusko	31/10/2009	4/12/2009	35	100.0
3	Implementation	10/11/2009	15/12/2009	36	100.0
3.1	Kusko Extension	10/11/2009	19/11/2009	10	100.0
3.2	New sources	15/11/2009	4/12/2009	20	100.0
3.3	Web Services	1/12/2009	7/12/2009	7	100.0
3.4	New/improve weight methods	1/12/2009	15/12/2009	15	100.0
4	Experimentation	1/12/2009	30/12/2009	30	100.0
5	Intermediate Report	15/12/2009	25/1/2010	42	100.0
6	Exams	25/1/2010	14/2/2010	21	100.0
7	Implementation	15/2/2010	7/6/2010	113	100.0
7.1	MTurk validation batch 1	15/2/2010	19/3/2010	33	100.0
7.2	Patterns exploration	1/3/2010	28/5/2010	89	100.0
7.3	Category Stability	24/3/2010	7/6/2010	76	100.0
7.4	MTurk Validation batch 2	30/3/2010	11/5/2010	43	100.0
8	Experimentation	1/3/2010	14/6/2010	106	100.0
9	Paper submission	15/6/2010	29/6/2010	15	100.0
10	Msc Thesis	30/6/2010	29/7/2010	30	100.0

Appendix B

MTurk Charts

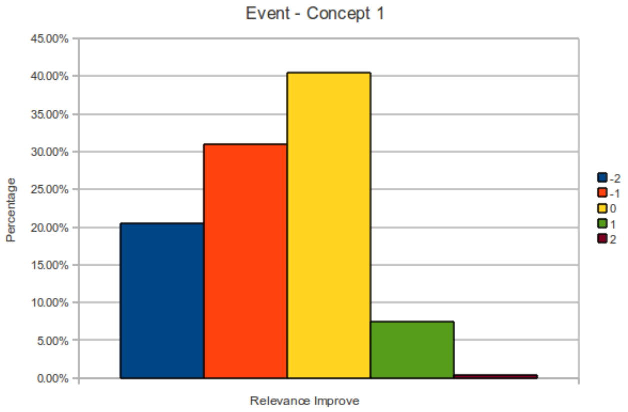


Figure B.1: MTurk - Events Batch 2 - Concept 1.

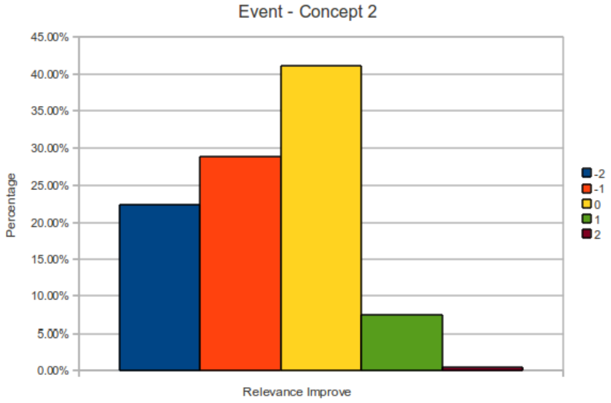


Figure B.2: MTurk - Events Batch 2 - Concept 2.

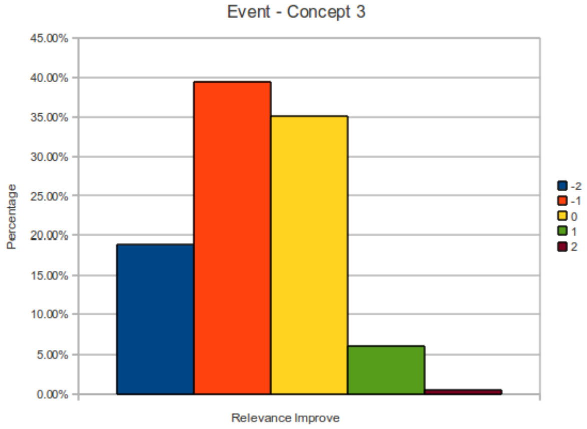


Figure B.3: MTurk - Events Batch 2 - Concept 3.

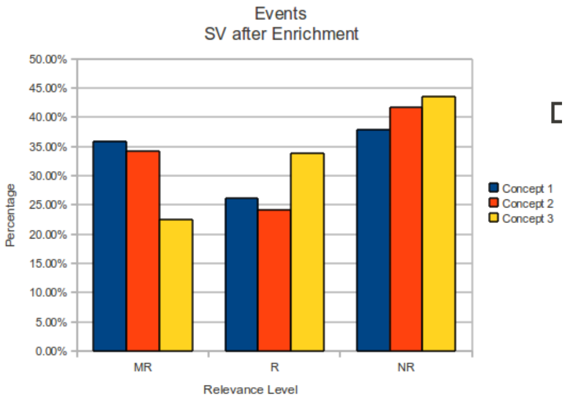


Figure B.4: MTurk - Events Batch 2 - After Enrichment.

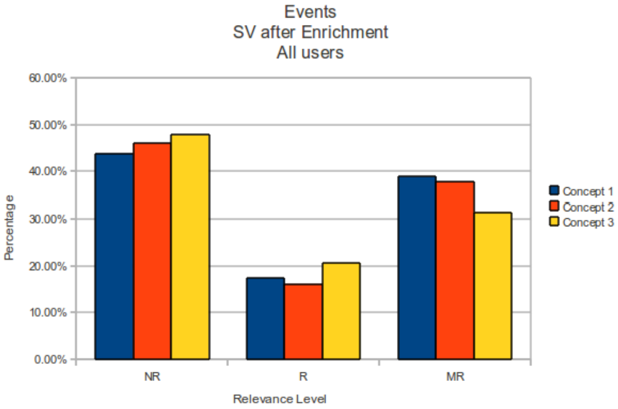


Figure B.5: MTurk - Events Batch 2 - After Enrichment with all users.

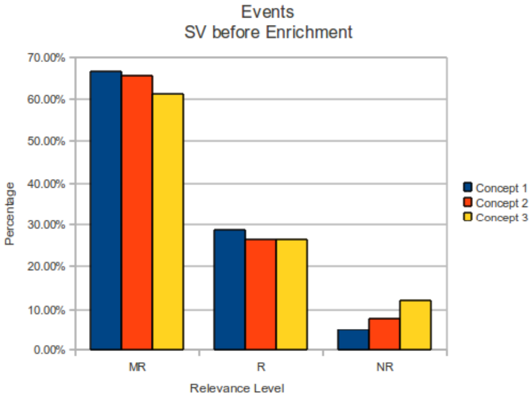


Figure B.6: MTurk - Events Batch 2 - Before Enrichment.

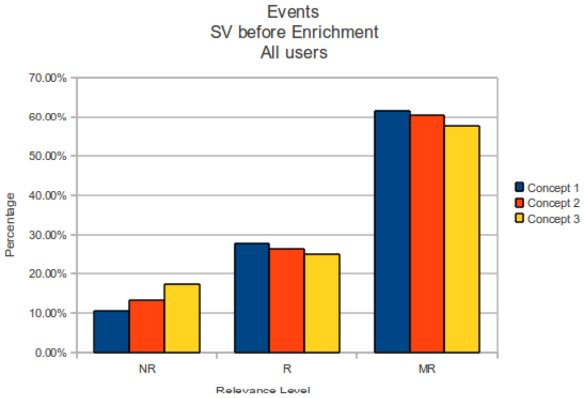


Figure B.7: MTurk - Events Batch 2 - Before Enrichment with all users.

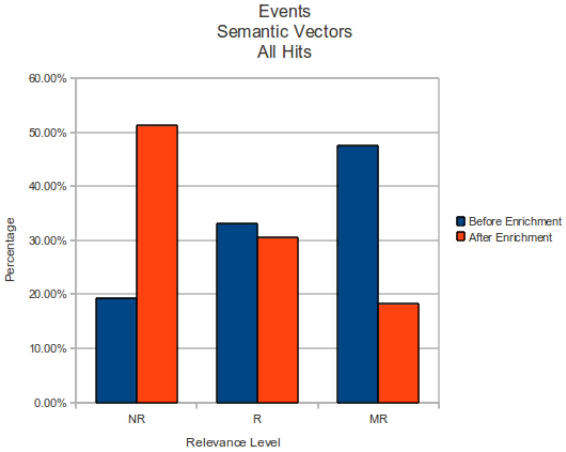


Figure B.8: MTurk - Events Batch 1 with all users.

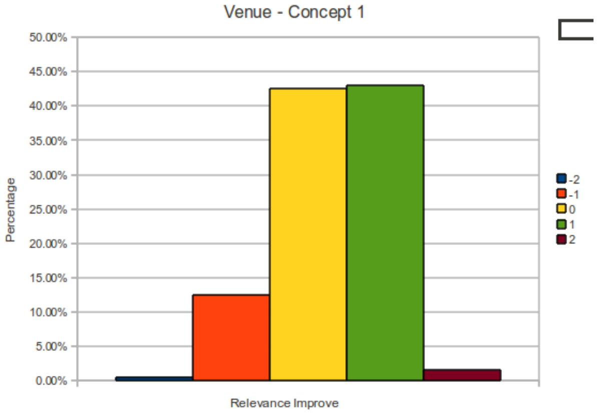


Figure B.9: MTurk - venues Batch 2 - Concept 1.

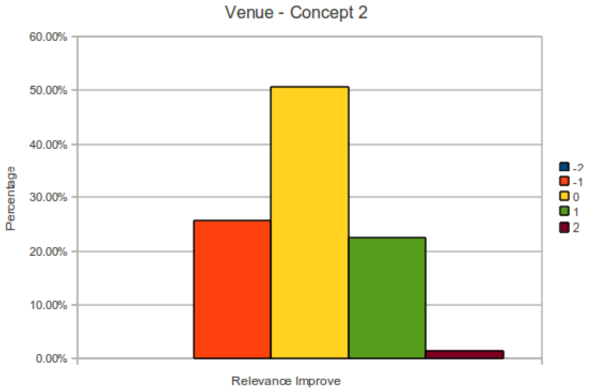


Figure B.10: MTurk - venues Batch 2 - Concept 2.

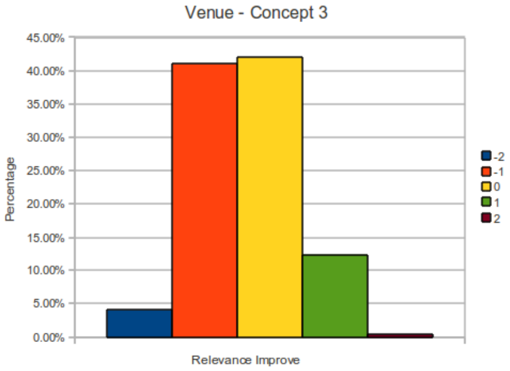


Figure B.11: MTurk - venues Batch 2 - Concept 3.

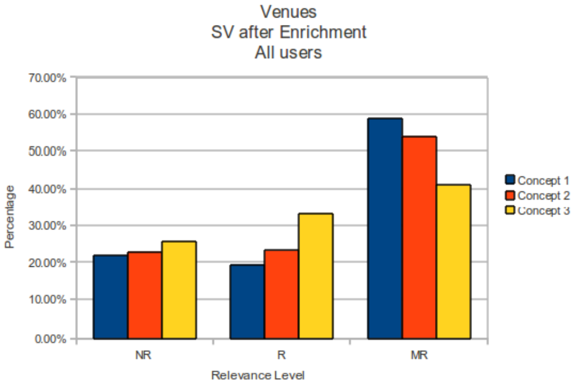


Figure B.12: MTurk - venues Batch 2 - After Enrichment with all users.

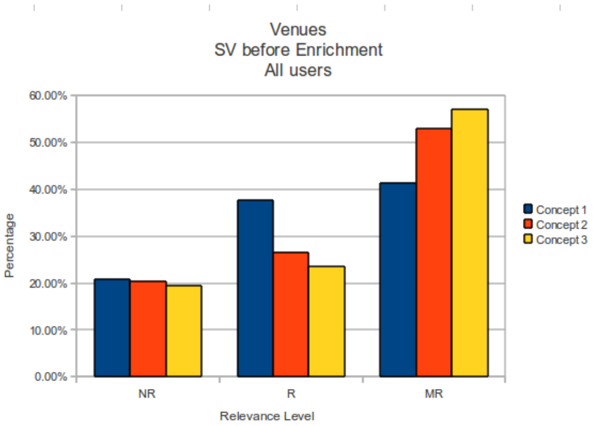


Figure B.13: MTurk - venues Batch 2 - Before Enrichment with all users.

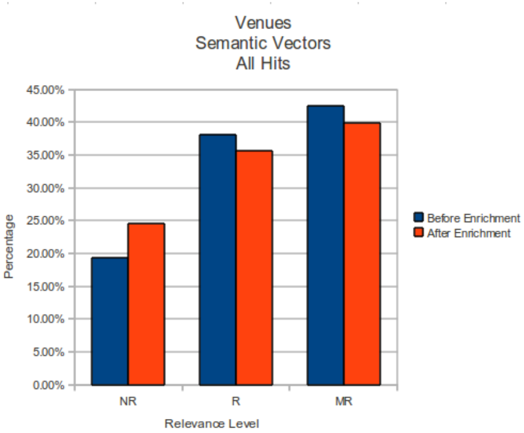


Figure B.14: MTurk - venues Batch 1 with all users.