SEMANTIC INDEXING OF WEB PAGES VIA PROBABILISTIC METHODS

In Search of Semantics Project

Fabio Clarizia, Francesco Colace, Massimo De Santo and Paolo Napoletano
Department of Information Engineering and Electrical Engineering, University of Salerno
Via Ponte Don Melillo 1, 84084 Fisciano, Italy

Keywords: Semantic index, Information Retrieval, Web Search Engine, Latent Dirichlet Allocation.

Abstract: In this paper we address the problem of modeling large collections of data, namely web pages by exploiting jointly traditional information retrieval techniques with probabilistic ones in order to find semantic descriptions for the collections. This novel technique is embedded in a real Web Search Engine in order to provide semantics functionalities, as prediction of words related to a single term query. Experiments on different small domains (web repositories) are presented and discussed.

1 INTRODUCTION

Modern search engines rely on keyword matching and link structure (cfr. Google and its Page Rank algorithm (Brin, 1998)), but the semantic gap is still not bridged.

The semantics of a web page is defined by its content and context, understanding of textual documents is yet beyond the capability of todays artificial intelligence techniques, and the many multimedia features of a web page make the extraction and representation of its semantics even more difficult. As well known any writing process can be thought as a process of communication where the main actor, namely the writer, encode his intentions through the language. Therefore the language can be considered as a code that conveys what we can call "meaning" to the reader that performs a process for decoding it. Unfortunately, due to the accidental imperfections of human languages, contingent imperfections may occurs then both encoding and decoding processes are corrupted by "noise", are ambiguous in practice.

We argue that the meaning is never fully present in a sign, but it is the limit point of a temporal, situated process, in which the text acts as a boundary conditions and in which the user is the protagonist. Following these claims we argue that semantic discovering and its representation could emerge through the interaction of facets, texts and users, that we call light and deep semantics.

In this direction Semantic Web (Berners-Lee et al., 2001) and Knowledge Engineering communities are both confronted with the endeavor to design different tools and languages for describing semantics in order to avoid the ambiguity of the encoding/decoding process. In the light of this discussions specific language has been introduced, RDF (Resource Description Framework), OWL (Ontology Web Language), etc., to support the creator (writer) of documents in describing semantic relations between concept/words, namely the metadata of the documents. During such a process of creation all the elements of ambiguity should be avoided because of use of a shared knowledge based on ontology as mean for semantics representation.

As a consequence the Web should be entirely rewritten in order to semantically arrange the content of each web pages, but this process can not be still realized, due to the huge amount of existent data and absence of definitive tools for managing and manipulating those languages. In the meantime, waiting for the semantic web starting, we could design tools for automatically revealing and managing semantics of the previous web by using methods and tools that don't ground on any web semantic specification.

In this direction, this paper deals with the problem of modeling large collections of data, namely web pages by exploiting jointly traditional information retrieval techniques with probabilistic ones in order to find semantic descriptions for the collections. This novel technique is embedded in a real Web Search Engine, in order to provide semantics functionalities,

Clarizia F., Colace F., De Santo M. and Napoletano P. (2009).
SEMANTIC INDEXING OF WEB PAGES VIA PROBABILISTIC METHODS - In Search of Semantics Project.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 134-140
DOI: 10.5220/0002010401340140

as prediction of words related to a single term query. Experiments on different small domains (web repositories) are presented and discussed.

The paper is organized as follows. In Section 2 we introduce basic notions about traditional and probabilistic indexing techniques. A probabilistic model, namely the topic model, is presented in Section 3 where a procedure for single and multi-words prediction is presented. An algorithm for building a semantic indexing is illustrated in Section 4 where illustrative examples of real environment are provided. Finally, in Section 5 we present some conclusions.

2 FROM TRADITIONAL TO PROBABILISTIC INDEXING TECHNIQUES

Several proposal have been made by researchers for the information retrieval (IR) problem (R. and Ribeiro-Neto, 1999). The basic methodology proposed by IR researchers for text corpora - a methodology successfully deployed in modern Internet search engines - reduces each document in the corpus to a vector of real numbers, each of which represents ratios of counts. Following this methodology we obtain the popular term frequencyinverse document frequency (*tf-idf*) scheme (Salton and McGill, 1983), where a basic vocabulary of "words" or "terms" is chosen, and, for each document in the corpus, a count is formed of the number of occurrences of each word. After suitable normalization, suitable comparison between term frequency count and inverse document frequency count, we obtain the term-by-document matrix W whose columns contain the *tf-idf* values for each of the documents in the corpus.

Thus the *tf-idf* schema reduces documents of arbitrary length to fixed-length lists of numbers, and it also provides a relatively small amount of reduction in description length and reveals little in the way of inter- or intradocument statistical structure. The latent semantic indexing (LSI) (Deerwester et al., 1990) technique has been proposed in order to address these shortcomings. Such method uses a singular value decomposition of the W matrix to identify a linear subspace in the space of *tf-idf* features that captures most of the variance in the collection. This approach can achieve significant compression in large collections.

Moreover, a significant step forward a full probabilistic approach to dimensionality reduction techniques was made by Hofmann (Hofmann, 1999), who presented the probabilistic LSI (pLSI) model, also known as the aspect model, as an alternative to LSI. The pLSI approach models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of "topics". Thus each word is generated from a single topic, and different words in a document may be generated from different topics. Each document is represented as a list of mixing proportions for these mixture components and thereby reduced to a probability distribution on a fixed set of topics. This distribution is the reduced description associated with the document.

While Hofmanns work is a useful step toward probabilistic modeling of text, it is incomplete in that it provides no probabilistic model at the level of documents leading to several problems: overfitting and probability assignment to a document outside of the training set is unclear. In order to overcome these problems a new probabilistic method has been introduced, called Latent Dirichlet Allocation (LDA) (Blei et al., 2003) that we exploit in this paper in order to catch essential statistical relationships between words contained in web pages' index. This method is based on the *bag-of-words* assumption - that the order of words in a document can be neglected. In the language of probability theory, this is an assumption of exchangeability for the words in a document (Aldous, 1985), which holds also for documents; the specific ordering of the documents in a corpus can also be neglected. A classic representation theorem establishes that any collection of exchangeable random variables has a representation as a mixture distribution - in general an infinite mixture. Thus, if we wish to consider exchangeable representations for documents and words, we need to consider mixture models that capture the exchangeability of both words and documents. In this paper we propose an hybrid proposal where the LDA technique is embedded in a traditional technique procedure, the *tf-idf* schema. More details are discussed next.

3 PROBABILISTIC TOPIC MODEL: LDA MODEL

As discussed before a variety of probabilistic topic models have been used to analyze the content of documents and the meaning of words. These models all use the same fundamental idea that a document is a mixture of topics but make slightly different statistical assumptions. In this paper we use the topic model, discussed in (T. L. Griffiths, 2007) based on the LDA algorithm (Blei et al., 2003), where statistic dependence among words is assumed. By following this approach, 4 problems have to be solved: *word*

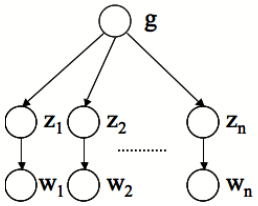


Figure 1: Graphical Models (T. L. Griffiths, 2007) relying on Latent Dirichlet allocation (Blei et al., 2003). Such Graphical Models (GM) don't allow relations among words by assuming statistical independence among variables.

patching, *prediction*, *disambiguation* and *gist extraction*, resulting in the GM reported in Figure 1.

Assume we have seen a sequence of words $\mathbf{w} = (w_1, \ldots, w_n)$. These n words manifest some latent semantic structure l. We will assume that l consists of the gist of that sequence of words g and the sense or meaning of each word, $\mathbf{z} = (z_1, \ldots, z_n)$, so $l = (\mathbf{z}, \mathbf{g})$. We can now formalize the four problems identified in the previous section:

- Word patching: Compute (w_i, w_j) from $\mathbf{w}$.
- Prediction: Predict $w_{n+1}$from $\mathbf{w}$.
- Disambiguation: Infer $\mathbf{z}$ from $\mathbf{w}$.
- Gist extraction: Infer $\mathbf{g}$ from $\mathbf{w}$.

Each of these problems can be formulated as a statistical problem. In this model, latent structure generates an observed sequence of words $\mathbf{w} = (w_1, \ldots, w_n)$. This relationship is illustrated using graphical model notation (Bishop, 2006). Graphical models provide an efficient and intuitive method of illustrating structured probability distributions. In a graphical model, a distribution is associated with a graph in which nodes are random variables and edges indicate dependence. Unlike artificial neural networks, in which a node typically indicates a single unidimensional variable, the variables associated with nodes can be arbitrarily complex. The graphical model shown in Figure 1 is a directed graphical model, with arrows indicating the direction of the relationship among the variables. The graphical model shown in the figure indicates that words are generated by first sampling a latent structure, l, from a distribution over latent structures, $P(l)$, and then sampling a sequence of words, $\mathbf{w}$, conditioned on that structure from a distribution $P(\mathbf{w}|l)$. The process of choosing each variable from a distribution conditioned on its parents defines a joint distribution over observed data and latent structures. In the generative model shown in Figure 1, this joint distribution is $P(\mathbf{w}, l) = P(\mathbf{w}|l)P(l)$. With an appropriate choice of l, this joint distribution can be used to solve the problems of word patching, prediction, disambiguation, and gist extraction identified above. In particular, the probability of the latent structure l given the sequence of words $\mathbf{w}$ can be computed by applying Bayes's rule:

$$P(l|\mathbf{w}) = \frac{P(\mathbf{w}|l)P(l)}{P(\mathbf{w})} \quad (1)$$

where

$$P(\mathbf{w}) = \sum_{l} P(\mathbf{w}, l)P(l) \quad (2)$$

This Bayesian inference involves computing a probability that goes against the direction of the arrows in the graphical model, inverting the generative process.

Equation 2 provides the foundation for solving the problems of word patching, prediction, disambiguation, and gist extraction.

Summing up:

- Word patching

$$P(w_i, w_j) = \sum_{\mathbf{w}-(w_i,w_j)} \sum_{l} P(\mathbf{w}, l)P(l) \quad (3)$$

- Prediction

$$P(w_{n+1}, \mathbf{w}) = \sum_{l} P(w_{n+1}|l, \mathbf{w})P(l|\mathbf{w}) \quad (4)$$

- Disambiguation

$$P(\mathbf{z}|\mathbf{w}) = \sum_{g} P(l|\mathbf{w}) \quad (5)$$

- Gist extraction

$$P(\mathbf{g}|\mathbf{w}) = \sum_{z} P(l|\mathbf{w}) \quad (6)$$

We will use a generative model introduced by Blei et al. (Blei et al., 2003) called latent Dirichlet allocation. In this model, the multinomial distribution representing the gist is drawn from a Dirichlet distribution, a standard probability distribution over multinomials, e.g., (Gelman et al., 1995). The results of LDA algorithm are two matrix:

1. the words-topics matrix Φ: it contains the probability that word w is assigned to topic j;
2. the topics-documents matrix Θ: contains the probability that a topic j is assigned to some word token in document d.

3.1 Single and Multi-words Prediction

Once we have the LDA computation for the index, we can compute predictions and semantic relations between documents.

As reported in (T. L. Griffiths, 2007) we need the single topic assumption for word prediction, namely $z_i = z$ for all i. This single topic assumption makes the mathematics straightforward and is a reasonable working assumption for this real application.

This also suggests a natural measure of semantic association, $P(w_i|w_j)$, in practice, given the word w_j (for a real IR environment it could be a single term query) we compute the probability to predict the word w_i. More in general we have:

$$P(w_{n+1}|w_n) = \sum_z P(w_{n+1}|z)P(z|w_n) \qquad (7)$$

Starting from the single word prediction we could generalize and compute the multi-words prediction, namely:

$$P(w_{n+m}, \cdots, w_{n+1}|w_n) = \qquad (8)$$

$$\sum_z P(w_{n+m}, \cdots, w_{n+1}|z)P(z|w_n) \qquad (9)$$

where m represents the number of words to be predicted. Each IR system performs term query functionalities that, due the nature of language is ambiguos, could not satisfy user intentions. A kind of single or multi-words prediction could be useful in order to aid the user to better perform his request.

4 SEMANTIC INDEXING FOR A REAL ENVIRONMENT

We propose a new indexing technique that, exploiting the topic model, reveals topics and semantic relations between words for the corpora. The index of this web search engine is composed of the traditional term-by-document matrix W whose columns contain the *tf-idf* values and the Θ and Φ matrix that are useful to compute word prediction. In Fig. 2 is reported a diagram for summarize this indexing procedure.

The probabilistic topic model is embedded in a real web search engine developed at University of Salerno and reachable through the URL http://193.205.164.64/isos after a registration procedure. Such a web search engine is part of a research project called *in Search of Semantics* (*iSoS*) which aims to develop a framework for extracting/revealing, representing and managing semantics of each kind of documents - text, web pages etc.

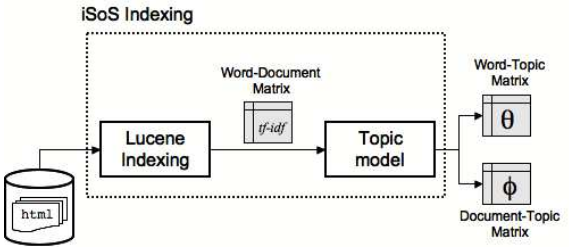


Figure 2: in Search of Semantic indexing procedure.

Figure 3: in Search of Semantic web search engine screenshot.

The project aims at investigating how *light* and *deep semantics* -and their mutual interaction - can be modeled through probabilistic models of language and through probabilistic models of human behaviors (e.g., while reading and navigating Web pages), respectively, in the common framework of most recent techniques from machine learning, statistics, information retrieval, and computational linguistics. In Figure 3 is showed a screenshot for the iSoS web search engine and in following we describe its principal functionalities.

4.1 in Search of Semantics: Functionalities

As discussed above, iSoS is a web search engine with advanced functionalities. This engine is a web based application, entirely written in Java programming language and Java Script Language embedding some of the open source search engine Lucene [1] functionalities. As basic functionalities it performs sintax query-

[1] http://lucene.apache.org/

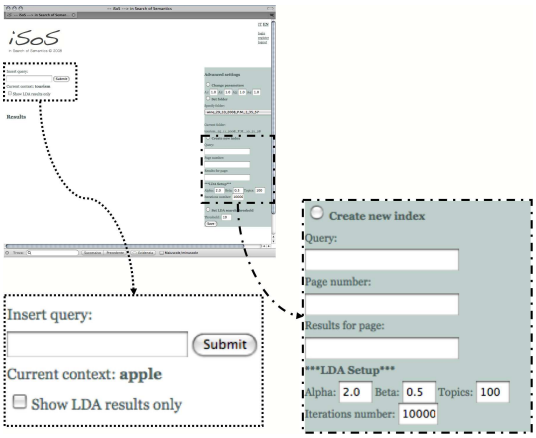


Figure 4: in Search of Semantic web search engine's functionalities screenshot.

ing, see the left side of Figure 4, and it gives results as a list of web pages ordered by frequency of the term query.

The iSoS engine operates, in the following order: 1. Web crawling, 2. Indexing, 3. Searching. Each web search engines work by storing information about web pages, which are retrieved by a Web crawler, a program which follows every link on the web. In order to better evaluate the performance of such web search engine, a small real environment is created. It performs a simplified crawling stage by submitting a query to a famous web search engine Google (www.google.com), and crawling the URL of the web pages contained in the list of results of Google. In Fig. 5 we report the code for the crawling stage.

During the indexing stage each page is indexed by performing the semantics indexing process discussed above.

The searching stage is composed of 2 main parts. The first is a language parsing stage for the query, where stop words like "as", "of " and "in", are removed and the second is a term searching stage in the *tf-idf* schema. During this stage the words related to the term query are predicted by using the Φ matrix.

4.2 Experimental Results

In order to show how the topic model is able to revealing semantics, we have indexed several web domain: *apple*, *bass* and *piano*. For each domain we have created a small web pages repository composed of 200 documents obtained by using the crawling procedure discussed above, namely by referring to Fig. 4 for the query *apple* we have:

```
query=apple, step=2, start=100
```

In the following we report result for the semantic indexing and for the multi-word prediction we have $m = 6$ for each domain. We used a java implementation of the LDA algorithm based on Gibbs sampling and for all the experiments we used 50 topics.

4.2.1 *Apple* Domain

In Fig. 6 we show some list of words extracted from the words-topics matrix Φ ordered by probability of belonging to such topic. The lists are truncated to the first 10 and then most probable words.

In Fig. 7 we show some multi-word prediction processes obtained by submitting several query to the iSoS web search engine: *macbook* , *tree* . We note that for *macbook* word the semantic index gives really related words belonging to the Apple Inc. domain. For what concerns the query *tree* we have words related to apple fruit domain.

4.2.2 *Bass* Domain

In Fig. 8 we show some list of words extracted from the words-topics matrix Φ ordered by probability of belonging to such topic. .

In Fig. 9 we show some multi-word prediction processes obtained by submitting several query to the iSoS web search engine: *fish* , *instruments* . We note that for *fish* word the semantic index gives really related words belonging to the sea bass domain. For what concerns the query *instruments* we have words related to instruments domain.

4.2.3 *Piano* Domain

In Fig. 10 we show some list of words extracted from the words-topics matrix Φ ordered by probability of belonging to such topic. .

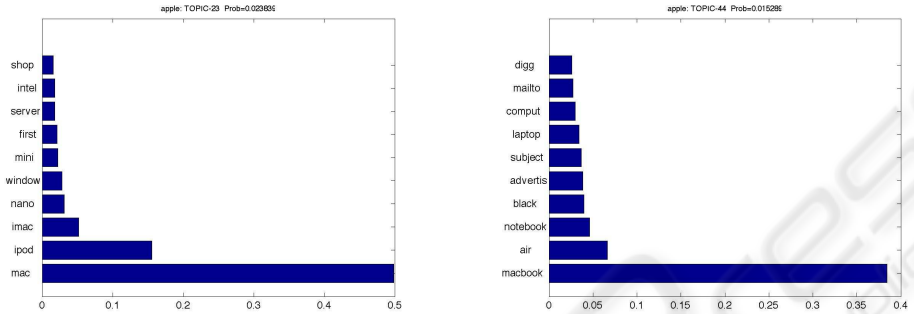
In Fig. 11 we show some multi-word prediction processes obtained by submitting several query to the iSoS web search engine: *architect* , *piano* . We note that for *architect* word the semantic index gives really related words belonging to the Renzo Piano Architect domain. For what concerns the query *instruments* we have words related to instruments domain.

5 CONCLUSIONS

In this work we presented a novel technique for indexing web pages based on a combination of traditional and probabilistic method, the topic model. We have experimented the proposed method in a real environment, a web search engine, namely 3 web domain for both the topics revealing and multi-word prediction

```
....
Socket s = new Socket("www.google.com",80);
PrintStream p = new PrintStream(s.getOutputStream());
p.print("GET /search?q="+query+"&num="+step+"&hl="+lan+"&start="+start+"&sa=N HTTP/1.0\r\n");
p.print("User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; rv:1.8.1) Gecko/20061010 Firefox/2.0\r\n");
p.print("Connection: close\r\n\r\n");
InputStreamReader in = new InputStreamReader(s.getInputStream());
BufferedReader buffer = new BufferedReader(in);
....
```

Figure 5: in Search of Semantic web search engine's crawler.

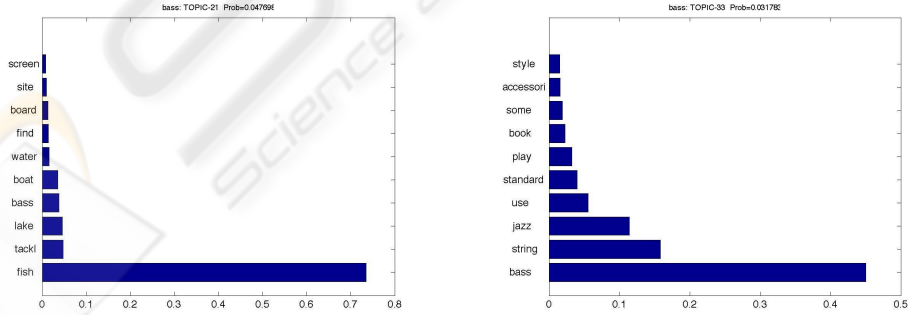


(a) (b)

Figure 6: Apple domain topics example. 6(a) *Computer shop* topic.6(b) *Apple Inc.* topic.

(a) (b)

Figure 7: Apple domain prediction. 7(a) query word *macbook*.7(b) query word *tree*.



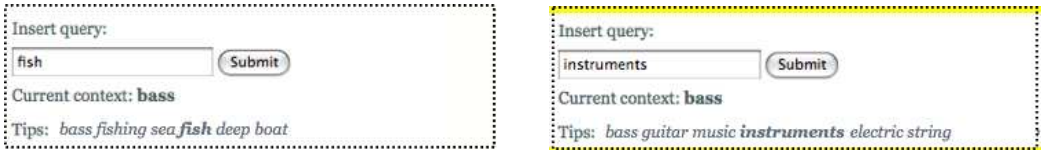
(a) (b)

Figure 8: Bass domain topics example. 8(a) *Sea bass* topic.8(b) *Bass instrument* topic.

tasks. The experiments confirm that such semantic indexing teachnique reveals semantic relations among words belonging to the same topic.

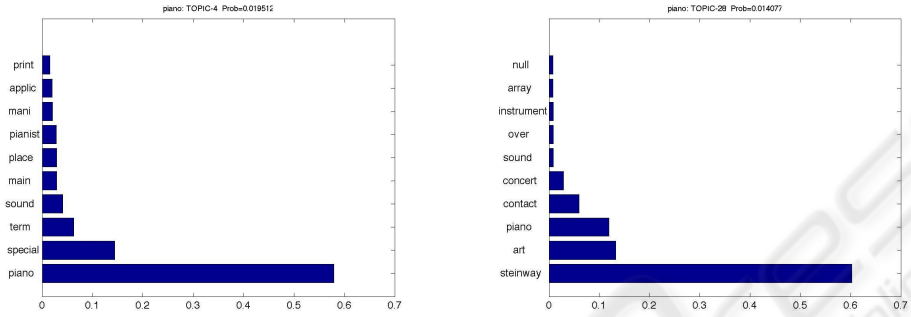
ACKNOWLEDGEMENTS

The authors wish to thank Luca Greco because part of this work is developed during his Master thesis in



(a) (b)

Figure 9: Bass domain prediction. 9(a) query word *fish*.9(b) query word *instruments*.



(a) (b)

Figure 10: Piano domain topics example. 10(a) *Piano instrument* topic.10(b) *Concert* topic.

(a) (b)

Figure 11: Piano domain prediction. 11(a) query word *architect*.11(b) query word *piano*.

Electronic Engineering at University of Salerno, supervised by Prof. Massimo De Santo and Dr. Paolo Napoletano.

REFERENCES

Aldous, D. (1985). Exchangeability and related topics. In Springer, B., editor, *École d'ete de probabilites de Saint-Flour XIII— 1983*, pages 1–198.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, May.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(993–1022).

Brin, S. (1998). The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117.

Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian data analysis*. New York: Chapman & Hall.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the Twenty-Second Annual International SIGIR Conference*.

R., B.-Y. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press, New York.

Salton, G. and McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill.

T. L. Griffiths, M. Steyvers, J. B. T. (2007). Topics in semantic representation. *Psychological Review*, 114(2):211–244.