E-Learning at Work and the Workplace
From Education to Employment and Meaningful Work with ICTs
Proceedings of the European Distance and E-Learning Network 2014 Annual Conference
Zagreb, 10-13 June, 2014
ISBN 978-963-89559-7-5
ISSN: 2707-2819
doi: https://doi.org/10.38069/edenconf-2014-ac-0031

ORBIS DICTUS – FROM LEXICAL STATISTICAL COMPUTATION TO NATURAL LANGUAGE PROCESSING AND SELF CUSTOMISATION

Nader A. M. Harb, Francesco Agrusti, Università degli Studi Roma Tre, Italy

Introduction

One of the main objectives of the Italian national funded project Adaptive Message Learning (am-Learning) is to produce a system capable of understanding the learner's lexical needs and providing him/her with the adapted study material that he/she will understand with the least help from a third party knowledge source (e.g. dictionaries, web, etc.). During the four years project, this result was obtained using lexical statistical computation techniques to measure relevant educational competences such as reading comprehension. This approach led us to manipulate the idea of "word frequency" as related to the difficulty of comprehension of a word in a certain document. In other words, the concept of frequency can be defined as how many times a word occurs in a huge collection of texts that fall under the same category (e.g. cardiology, physiotherapy, sociology, etc.) – called *corpus*. The higher the frequency the word holds, the higher the chance (probability) that the reader (student) already knows it (understanding its meaning in the document), and vice-versa.

One of the product of the am-Learning project is an advanced e-learning platform (or LMS – learning management system) called Orbis Dictus. It is already working and it implements the above approach to deliver an automatically adapted e-learning materials and tests based on lexical statistical algorithms. This innovative platform is formed by three distinct technological tools, each of them devote to provide different LMS functionalities: the *LexMeter* module outlines an initial user profile assessing the learner's characteristics in terms of his/her lexical competency; the *ProgressMeter* module creates short *cloze* tests to report and monitor the learner's gradual improvement through the learning path; using the results obtained by the other two, the third module, called *Adapter*, automatically adjusts the text document (e.g. manuals) in accordance to the follow hypothesis: introducing more detailed explanation of low-frequency (hard) words helps students better understand the given text material. In other words, starting from a fixed text inserted by the course tutor, the study material is automatically integrated with definitions and explanations in order to provide the student with an already tuned text, matching his/her reading skill.

Some enhancements were proposed to improve this approach taking advantage of the vast collection of words added daily to the web by its users, collecting new data with every second passing, in addition to building a system that will not need to be explicitly programmed every time new data is added, or every time the system makes a wrong judgement or a right one, but simply learns new knowledge from the newly collected data (Machine Learning).

One more addition deserves mentioning is the natural language processing, in this field some studies and algorithms were adopted in order to help the system consider the articles and books not as a collection of single words, but as a collection of phrases, paragraphs that cover a particular subject (Aboutness concept).

Web crawling

Using the XSL/XSLT language, a number of web crawlers has been developed, each of which visits a specified website, analyses it and stores all the links in the website, whether they point to an internal page in the same website or an external link, after analysing the first page, the crawler comes back with the information and the set of links found, copies itself as many times as the links found in the initial page, starts the same process with the links set one by one, finding new content, new links to follow and so on.

Web crawlers have been developed and spread over the web to find Italian language websites that contain articles, news or discussion spaces. As for now, 18 Italian language websites has been found, analysed and continuously monitored for updates. One example is http://it.docsity.com that has more than 11,000,000 question pages, each page contains one question made by a user (university student) and many answers with a voted best answer reply, In addition to 300,000 presentations and notes made on didactical materials used in Italian universities. This approach goes as long as there new content added to the crawled websites, in addition to the possibility to add new websites to analyse. The second step starts after the crawling process ends, cleaning the text and removing the noise that would only confuse the collected data (Latent Semantic Analysis).

Collecting the data from the web helps facing the challenge of updating the De Mauro's VDB, whether the update includes adding new common words, deleting old unused ones and updating the words ratings (words frequency/occurrence)[1]. And now with words being continuously retrieved, another approach is needed in order to understand the meaning behind not only the words, but of the sentence, context and the whole document.

Machine learning

Machine learning follows the same concepts of the human learning, but with a mathematical basis of a specific problem domain that are not seen as the way humans learn, but the outcome is the same however, a system developed and built following the machine concepts will improve the output quality with trial-and-error, training and the usage to previous data.

[1] Giuliani et al. (2005) demonstrated that only a small part of the third list of De Mauro's VDB (Alta disponibilità) is used these days.

Machine learning applications vary from building autopilots, handwriting recognition (optical character recognition), computer vision to database mining (which was adopted in order to categorise and arrange collected web data in our case) and natural language processing (NLP).

Latent semantic analysis (LSA)

Latent Semantic Indexing (also known as Latent Semantic Analysis) can be defined as a technique to discover the underlying meaning or concept (main idea) of a document. If every word held only one meaning, then LSA would have been easy to achieve (Figure 1).

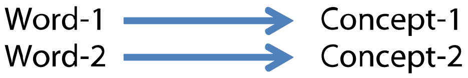

Figure 1.

Unfortunately, languages hold many synonyms, words with multiple meanings, and some obscurities that make it sometimes hard even for people to understand (Figure 2).

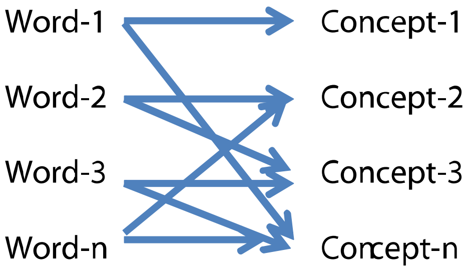

Figure 2.

One more problem should be addressed here, which is the fact that authors have a large choice of words for them to choose from, this makes it harder to analyse. This casual choice of words from authors leads to noise in the concept that describes the relationship between words and meanings. LSA proposes some simplifications in order to face and eventually solve this problem:

- Documents are considered as a "box of words" where the order of appearance is ignored.
- Concepts can be found usually from patterns of words that usually are together in a document (carta di credito, conto, bilancia, could possibly occur in a document about banking & finance).
- Words used in the same context usually hold the same meaning.

The LSA implementation has been made throughout the use of python programming language, using the python numerical library NumPy that will help with the creation of the count matrix, and python's scientific library SciPy that includes the Singular Value Decomposition (SVD).

Stop words (words that do not contribute to the meaning such as: e, il, la, in, per, a … etc.) and ignore characters sets (&, $, ”, £, !, ”, ;, etc...) should be considered before parsing a document. The parser method splits the document into single words after removing/ignoring words and characters that match the characters and words inserted in the stop words and ignore characters lists.

After parsing all the documents, the words[2] (dictionary keys) that occur in more than one document are exported and arranged, and a matrix is built with rows representing the words, and columns representing the documents parsed, in the end, for each document and dictionary key (word) pair, the corresponding matrix cell is incremented.

Table 1: shows an example matrix

	Documents								
Words	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**Dn**
docimologia		1	1	1	3		1	1	
didattica	1				2	2	2		1
valutazione		1		1	1		4	1	
procedure	1	1	1	1	1	1	2	1	1
decisione					1			1	
votazione	1	1		1		4	2	1	1

In complex LSA/LSI systems, the previous matrix, which can also be referred to as raw matrix is often processed and modified in order to distinguish easy from hard words (common from specific / technical), in a simplified manner, a word that occurs in nearly all the documents should have lesser weight than the word that occurs in only one document, one well-known method is the term frequency – inverse term frequency, which can be achieved by applying the following formula to the previously calculated counts in the count matrix:

TFx,y = (Nx,y / N*,y) * Log(D / Dx)[3]

After the TF formula has been applied to the matrix counts, an algorithm is adopted in order to examine and analyse the new count matrix called the Singular Value Decomposition algorithm (SVD). The motive it was considered as a useful addition to the whole approach is that it achieves the best possible reconstruction of the count matrix with the least possible information, it discards the noise, and emphasizes strong patterns and trends. The trick in using SVD is in figuring out how many dimensions or "concepts" to use when approximating the count matrix. Too few dimensions and important patterns are left out, too many and noise caused by random word choices will creep back in.

[2] Words are defined as dictionary keys in the python program built, which are the words that are considered as dictionary entries, those entries have values assigned to them based on their occurrence in texts collected (word frequency).

[3] Nx,y = the number of times word x appears in document y (the original cell count).
N*,y = the number of total words in document y (just add the counts in column y).
D = the number of documents (the number of columns).
Dx = the number of documents in which word x appears (the number of non-zero columns in row x). In this formula, words that concentrate in certain documents are emphasized (by the Nx,y / N*,y ratio) and words that only appear in a few documents are also emphasized (by the Log(D / Dx) term).

After the analysis is done, the system decides based on the correspondence and the amount of noise the webpage has in order to either include or exclude it in the final list of websites to retrieve data from continuously. So far, only 18 websites matched the criteria, which is not a small amount of initial data but rather a huge one (roughly 16,000 documents in addition to 900,000 discussions), and this is only one small advantage of using LSA techniques. The other and most fruitful advantage will be the update of De Mauro's VDB whenever a sufficient archive is created in order to reduce the error margin to a minimum (target is at least 25,000 documents)

The final objective was to have a system capable of profiling every student, holding vital information about the student that will help the system obtain a better understanding of that student. This does not only imply that the system will make decisions based on the data it has, but it will learn and deduce new knowledge about the user using the actual data in possession without being programmed explicitly, which is considered as the main concept behind Machine Learning and the reason why it was adopted.

Future developments

Having already implemented all that was mentioned and described beforehand, when trying to put everything together, a final step is yet to be made and developed, a user profile that holds not only basic information. Users and words should be considered as connected nodes, people with similar education and interests have highly similar lexicons, words should have some identifiable characteristics that can help connect similar words together (Bank, Account, Balance are connected to the finance domain), so, not only the word frequency should be considered, but a group of words frequencies should be considered as a set in order to find those connections. With this said, two new different approaches are needed: the former is a deeper semantic approach, in order to categorise and discover similarities between words; the latter is a new approach to define relations and similarities between users (students).

References

1. Bajwa, I.S. (2010). Context Based Meaning Extraction by Means of Markov Logic. In *International Journal of Computer Theory and Engineering IJCTE, 2(1),* (pp. 35-38).

2. Bird, S.; Klein, E. and Loper, E. (2009). *Natural Language Processing with Python.* O'Reilly.

3. Gouws, S.; Hovy, D. and Metzler, D. (2011). Unsupervised Mining of Lexical Variants from Noisy Text. In *Proceedings of the First workshop on Unsupervised Learning in NLP,* (pp. 82-90).

4. Giuliani, A.; Iacobini, C.; Thornton, A.M. (2005). La nozione di vocabolario di base alla luce della stratificazione diacronica del lessico dell'italiano. In T. De Mauro & I. Chiari (eds.), *Parole e numeri. Analisi quantitative dei fatti di lingua,* (pp. 193-213). Roma, Aracne.

5. Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. In *Machine Learning Journal, 42(1),* (pp. 177-196).