

PtiClic: A Game for Vocabulary Assessment combining JeuxDeMots and LSA

Mathieu Lafourcade¹ and Virginie Zampa²

¹ LIRMM-TAL, Univ. Montpellier 2 France
mathieu.lafourcade@lirmm.fr

² LIDILEM-DIP Univ. Grenoble 3 France
virginie.zampa@u-grenoble3.fr

Abstract. One interesting path for designing software for vocabulary acquisition and assessment could be games. In this article we present PtiClic, a lexical game based on the principles behind JeuxDeMots and combined with LSA. Such a game can foster the interest of young people in developing lexical skills in either a tutored or an open environment.

Keywords: Lexical acquisition, lexical network, serious games, LSA, JeuxDeMots

1 Introduction

Developing software for vocabulary acquisition and/or assessment in general, and for young people in particular, is a risky business. There are several difficulties. First, we have to be able to design an activity that can foster an interest in learning which is not easy in the case of children. Then, the underlying dictionaries or lexical databases can prove tremendously hard to develop, especially if we want to go beyond just a list of words with parts-of-speech and venture into the realm of lexical functions and the relations intertwining terms.

Automated acquisition of lexical or functional relations between terms is necessary in a large number of tasks in Natural Language Processing (NLP) outscoping largely Technology-Enhanced Learning of language. These relations that we find generally in thesauruses or ontologies can of course be revealed in a manual way; for example, one of the oldest thesauruses is Roget's, its current version being (Kipfer, 2001), or the most famous lexical network is Wordnet (Miller, 1990). Such relations can be also determined computationally from corpora of texts, for example (Robertson and Spark Jones, 1976), (Lapata and Keller, 2005), or (Landauer et al 1998) in which statistical studies on the distributions of words are made. Moreover, many applications of NLP require information of various natures, like synonymy or antonymy, but also relations of hyperonymy / hyponymy, holonymy / meronymy etc. The building of such relations, when done manually by experts, requires resources

which can be prohibitive, while their automatic extraction from a corpus can be biased by the chosen texts.

The method developed here relies on a contributory system, where the users feed the relation database through a game (JeuxDeMots). Furthermore, contrary to conventional methods which aim generally at static lexical information, the prototype introduced here is able to acquire evolving lexical information over time. From this game, we have designed a sequel, named PtiClic, which is a simpler version of JeuxDeMots aimed at vocabulary acquisition and assessment.

In this article, we first present the principles of JeuxDeMots¹ a game for building a lexical database containing various lexical functions. Secondly, we explain briefly how we produced a training corpus for LSA from the lexical network. Finally, the proposal of the PtiClic game is described with some insight into the rationale behind the design.

2 How to Build a Lexical Network through a Game

2.1 Principle of JeuxDeMot

To ensure a system leading to (quality and consistency in/the quality and consistency of) the lexical database, it has been decided that the relations anonymously given by a player should be validated by other players, also anonymously. In practical terms, this means a relation is considered valid if it is given by at least one pair of players. This process of validation is similar to the one used by (von Ahn and Dabbish, 2004) for the indexation of images or more recently by (Lieberman and al., 2007) to collect common sense knowledge. As far as we know, this has never been done in the field of lexical networks.

A game takes place between two players, in an asynchronous way, based on the concordance of their propositions. When a first player (A) begins a game, an instruction concerning a type of competence (synonyms, antonyms, domains ...) is displayed, as well as a term T randomly picked from a database of terms. This player A has then a limited timeframe in which to answer by giving propositions which, in his view, correspond to the instruction applied to the term T. The number of propositions which can be made is limited so that players do not just type anything as fast as possible, as we want the players to take time to think. The same term, along with the same instruction, is later proposed to another player (B); the process is then identical. To increase the playful aspect, for any common answer between the two players, they each receive a given number of points. The calculation of this number of points (explained in section 2.2) is crafted to induce both precision and recall in the feeding of the database.

¹ JeuxDeMots is available at <http://jeuxdemots.org>. An English version has recently been added, as well as a Thai version and a Japanese version (both in development), at <http://www.lirmm.fr/jeuxdemots/world-of-jeuxdemots.php>

For the target term T, we do not record answers given only by one of the two players but record the common answers given by player pairs. This allows the construction of a lexical network connecting the terms by typed and balanced relations, validated by pairs of players. These relations are labeled by the instruction given to players and they are weighted according to the number of pairs of players who proposed them. The structure of the lexical network relies on the notions of nodes and relations between nodes, as reminded by (Polguère, 2006). Every node of the network is constituted by a lexical item (term or expression) grouping together all its lexical items and the relations between nodes refer to lexical functions, as presented by (Mel'cuk and al., 1995). Nodes are constituted by an initial set of terms, but if both players in the same game suggest a term hitherto unknown (i.e not in the database), it is then added to the lexical database. Figure 1 presents the relations acquired for the French term *aille* (*wing*).

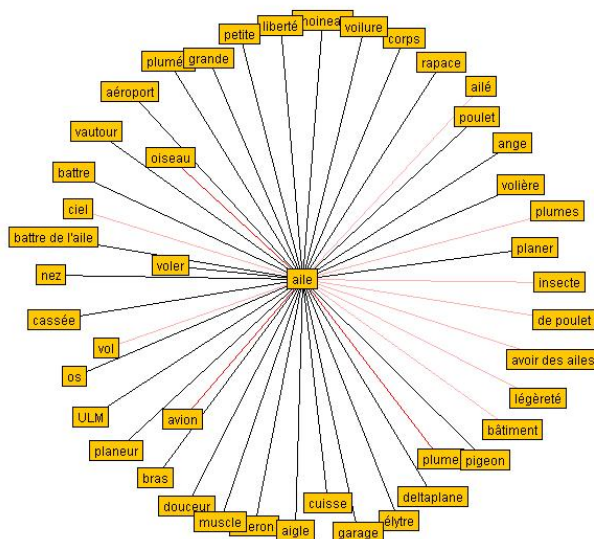


Fig. 1 Partial example of the lexical network for *aile*.

```

aile61 relations ==>
aile --r_assoc:370-> oiseau
aile --r_assoc:370-> voler
aile --r_assoc:340-> avion
aile --r_assoc:260-> plume
aile --r_assoc:140-> poulet
aile --r_assoc:130-> vol
aile --r_loc:130-> oiseau
aile --r_assoc:110-> cuisse
aile --r_assoc:100-> ange
aile --r_assoc:90-> planer
aile --r_loc:90-> avion
aile --r_holo:80-> avion
aile --r_holo:80-> oiseau
aile --r_assoc:70-> deltaplane
aile --r_assoc:60-> pigeon
aile --r_syn:60-> aileron
aile --r_syn:60-> bras
aile --r_has_part:60-> os
aile --r_has_part:60-> plume
aile --r_holo:60-> aigle
    
```

```

aile --r_loc:60-> ange
aile --r_loc:60-> bâtiment
aile --r_loc:60-> volière
aile --r_assoc:50-> ailé
aile --r_assoc:50-> battre
aile --r_assoc:50-> bras
aile --r_assoc:50-> bâtiment
aile --r_assoc:50-> insecte
aile --r_assoc:50-> moineau
aile --r_assoc:50-> planeur
aile --r_assoc:50-> plumes
aile --r_assoc:50-> voiture
aile --r_syn:50-> voilure
aile --r_syn:50-> élytre
aile --r_hypo:50-> plume
aile --r_has_part:50-> muscle
aile --r_holo:50-> ULM
aile --r_holo:50-> pigeon
aile --r_holo:50-> voiture
aile --r_loc:50-> aigle
aile --r_loc:50-> aéroport
aile --r_loc:50-> ciel
aile --r_loc:50-> garage
aile --r_loc:50-> pigeon
aile --r_loc:50-> poulet
aile --r_loc:50-> vautour
aile --r_loc:50-> voiture
aile --r_carac:50-> cassée
aile --r_carac:50-> grande
aile --r_carac:50-> petite

mouche --r_assoc:260-> aile
plume --r_assoc:240-> aile
oiseau --r_has_part:230-> aile
oiseau --r_assoc:200-> aile
insecte --r_has_part:150-> aile
poulet --r_assoc:150-> aile
rapace --r_has_part:140-> aile
volaille --r_has_part:140-> aile
papillon --r_assoc:130-> aile
avion --r_has_part:120-> aile
cuisse --r_assoc:100-> aile
nez --r_has_part:90-> aile
coq --r_has_part:80-> aile
avion --r_assoc:70-> aile
voler --r_instr:70-> aile
Ailette --r_assoc:60-> aile
faucon --r_has_part:60-> aile
frégate/oiseau --r_assoc:60-> aile
plume --r_holo:60-> aile
Icare --r_assoc:50-> aile
battement --r_assoc:50-> aile
carrosserie --r_has_part:50-> aile
deltaplane --r_has_part:50-> aile
fée --r_assoc:50-> aile
huîtrier-pie --r_assoc:50-> aile
poule --r_has_part:50-> aile
poule --r_assoc:50-> aile
poulet --r_hypo:50-> aile
toucan --r_assoc:50-> aile
voilure --r_syn:50-> aile
voler --r_assoc:50-> aile

```

aile **31 relations** <==

The JeuxDeMots software was mainly developed in PHP / MySQL; some secondary programs were written in the Java and C ++ languages. The user interface, the computation of scores, but also the notions of levels and points of honor, winning terms, trials between players etc., as well as the display of players' rankings, were implemented to make the game more attractive. The purpose is to incite players to return regularly to the site, and thus to increase the number of acquired relations accordingly: it is the major interest of this game compared to some other software programs which would merely ask users to provide relations making them certainly more aware of their role as experts, but probably leading them to spend less time on the game.

2.2 How is the Game Played?

Every time a player connects to the site and starts a game, an instruction is displayed for a few seconds (for example: "give ideas associated with ..."), before the term to which this instruction applies appears on the screen. This term is randomly picked from a base of about 150,000 terms. The player then has one minute to give their answers. If the player is (B), the result of the game, the suggestions made by player (A) and the number of points won are immediately displayed. If he is a player (A), the equivalent information will be sent to him by e-mail after (B) has played (this last point also serving as a reminder to return to the site and continue playing the game). The games proposed to the player are either "starter games" where he is a player (A), or "games to be completed" in which he is a player (B). Thus, there are a certain number of games to be finished.

For a given term and a given instruction, if a player has no idea, he can "pass", then ending the game prematurely. There are two main reasons for the player not to give any answer: either the term is not a common term (for example: "*gnomon*", which is an old Greek sundial), or the instruction applied to this term does not have any clear meaning (for example: "*contraires de pigeon ?*" ("*Opposites for pigeon ?*"). The system then records the fact that this term is may not have many lexical relations, in particular with regard to this instruction; consequently, the term coupled with this instruction will come up less often.

Any game created with a player (A) generates two games to be finished. Indeed, if it were not the case, player (B) would just have to pass without making any suggestion which may induce a feeling of frustration for player (A) who might lose interest in the game. This is why more games to be finished than beginning games are offered to players as it is indeed more rewarding to get the immediate result of their propositions.

In order to allow each player to compare themselves with the others, it is possible to display a summary table of the recorded players, with their performances. The member list is ordered according to their points of honor, as well as their best scores obtained in a game.

2.3 What are the Results?

The current version of JeuxDeMots is relatively recent: it was released in July 2007. In approximately twelve months, more than 1200 players have been registered and most of them connect several times a week. More than 120,000 games have been played: they have brought to the foreground more than 180,000 relations, among which 80,000 of the "associated ideas" type. At present, more than 2000 relations are taboo, i.e. they have been validated by a sufficient number of player pairs, and therefore to encourage the production of new links between words, players are informed that these words are "taboo" and will not win any points.). There is a fast emergence of the relations and we also note that statistically the strongest are created first (i.e. they are the most spontaneous ideas the player comes up with). The evolution of the base of terms is inevitably slower: to date, it amounts to

approximately 163,000 terms; players have already added more than 8000 new terms to it, mainly related to current events.

3 From a Lexical Network to a LSA Corpus

We use LSA to produce a cloud of words from a target word. From a given word, with LSA we can obtain any number of similar terms using a knn (K nearest neighbors) algorithm. The issue raised here is how to produce a corpus for training LSA from a lexical network? First, we have the following assumptions about using LSA:

- The context window is limited to a paragraph ;
- All words in a paragraph are equal in building the context of each of them.

3.1 From a Lexical Network to a Training Corpus for LSA

The solution we adopted, although imperfect, is quite straightforward. We enumerated all the associations contained in the database of JeuxDeMots and produced for each of them a duplicated number of paragraphs roughly proportional to the weight of the relation divided by 10. This approximation is motivated by the desire not to inflate the corpus too much. We did not, at this stage, take into account the type of relation. For example, for the relations presented above for *aille*, we obtain:

| | | |
|-----------------------|------------------------|-----------------------|
| aille | oiseau | (duplicated 37 times) |
| aille | voler | (duplicated 37 times) |
| aille | avion | (duplicated 34 times) |
| ... | | |
| aille | vol | (duplicated 13 times) |
| ... | | |
| aille | muscle | (duplicated 5 times) |
| aille | ULM | (duplicated 5 times) |
| aille | pigeon | (duplicated 5 times) |

In this way, we produce a corpus of 1.3 million paragraphs, each associating two terms. This corpus is given to LSA to compute a similarity matrix between all terms.

3.2 From a List of Similar Words to a Cloud

We intend to produce a cloud of n words (in the prototype of PtiClic we display between 20 to 30 terms). To do so, we ask LSA to produce a knn (k nearest neighbors) list from the similarity matrix with k equal $5*n$. We select randomly those terms among the k terms, in such a way as to produce different clouds for different games with the same target word.

3.3 Why use LSA?

At this point, one question may arise: why use LSA rather than directly producing the cloud of words from the lexical network? Using LSA on a corpus extracted from the lexical network has the following advantages:

- Once compiled, obtaining all distance neighbors and not only immediate neighbors is much more computationally efficient than using a graph walking. In this respect, the application of LSA can be viewed as a direct projection of the lexical network to a space;
- Using LSA we are both making symmetric the relations of the network and reducing the noise;
- The transitivity property of LSA allows catching terms that are not immediately related, or co-hyponyms, which is definitively interesting in a vocabulary assessment task.

3.4 Why use the JeuxDeMots Lexical Network?

Why did we not use a normal corpus of texts for training LSA? A first answer is that building a text corpus that is balanced between text genres is difficult. Secondly, although it is outside the precise topic of this paper, it is interesting to assess the transformation of the lexical network through LSA concerning term similarity. Some terms in the graph may be implicitly related, and LSA can “discover” those hidden relations. The PtiClic games can allow them to be validated.

5 Principle of PtiClic

The game PtiClic (<http://pticlic.org>) is just a derivation and simplification of JeuxDeMots. Here, we will firstly outline how it is played, and then explain the reasoning behind the choices made.

To begin, a first player (we called the Tutor) selects a term (the target) and the system proposes a cloud of words as produced by LSA. The Tutor then, selects some of these terms by clicking on them and, from a set of menus, builds the definition of the task to completed by the other players (known as Learners). The description of the task is basically to select a given number of terms related to the target term by a lexical function. For example, we can have the following tasks:

- Select up to 5 synonyms
- Select up to 3 most closely related terms
- Select up to 4 terms opposites (antonyms)
- Select up to 6 terms which are not part of the <target>
- Select 4 terms that are kinds of/a kind of <target>

The tutor can also produce some comments (as free text) that will be displayed at the end of the game.

When playing, the learners get the target word, the description of the task and the cloud of words. The player then can click on the words of the cloud to select or unselect them. When finished he can validate his/her answers. The result is then displayed, comparing answers proposed by the Tutor and the learners. Points are given to the learners on the basis of 2 for each proper answer and -1 for each wrong ones.

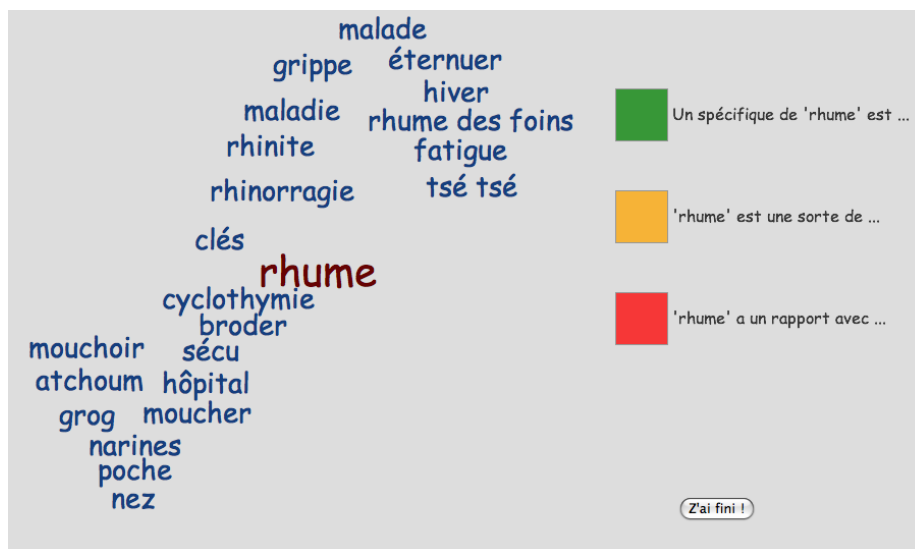


Fig. 2 Example of word cloud around the term “*rhume*” (cold N). Three drop-zones are present and correspond to “specific”, “kind-of” and “freely associated ideas”.

Although, for a given game built by the Tutor, the cloud of words remains the same, words are displayed in a random order to the Learners. Any given order (for example alphabetical) would produce a bias to the players. By randomly presenting words, we statistically reduce this bias.

PtiClic is a closed world contrary to JeuxDeMots which is an open game, as players can freely propose any terms. This design seems more adapted to new learners of the language, either young people or recent second-language learners. JeuxDeMots, as an open world, presents some obstacles like proper spelling for instance, and forces players to activate otherwise passive vocabulary which may be difficult for new learners, especially if they are indeed trying to acquire this vocabulary.

PtiClic, on the other hand, is simpler in its execution and contributes more to vocabulary acquisition and assessment. The game can be played during a strict time frame or without time limit. In the case of a session aimed more at vocabulary assessment, a time limit for each game is generally set.

A prototype version of PtiClic has been developed in a non-tutored environment. Anyone can play, not specifically people learning vocabulary. In this case, the creator is rewarded with the same amount of points as a player for a given game. This aspect

is quite significant, as the creator is induced to select proper lexical items according to the task.



Fig. 3 Example of result of the previous game. The players got 4 points. 7 words of the cloud were dropped on the proper zone, 2 words were not given (in grey) at all, and 4 words were wrong. We can notice that some words could have been dropped in several places. The score is computed in the more favorable way to the player.

At the time of writing of this paper, we do not have more than first impressions made on some users of PtiClic (mainly children from 8 to 14). Overall, the game was found to be very enjoyable and we discovered that it could lead to vocabulary acquisition (although at first the project was essentially aimed at assessment). Indeed, in games without a strict time frame (of around one minute), players went online for dictionaries and encyclopedia to gather information about words in the cloud they didn't know or were not sure about.

The game aspect - in particular the point system - made children try to score as much as possible, forgetting completely the learning aspect while doing so.

6 Conclusion

The JeuxDeMots prototype is an on-line game on the Web the objective of which is the construction of a lexical network. Making a game was justified by the assumption that it would attract a lot of people from various horizons. The emergence of labeled and weighted relations between terms is made through the gaming activity of a large number of users. These users are certainly not linguists, but we strongly believe that both their number and variety will allow obtaining a lexical network with a satisfactory coverage and precision for general knowledge. Our purpose is not the constitution of an experts' database, but rather the representation of common general knowledge.

From this lexical network, and thanks to LSA, we were able to produce the lexical data that serves as the foundation for another game: PtiClic. This game is a closed world, i.e. players have to make selections among proposals according to a given task, instead of proposing terms by themselves. This game seems to be more appropriate

for people still acquiring vocabulary (either young players, or second-language learners). The first experiments seem to confirm the idea that presenting a learning activity through a game with scores, involving emulation between players, is an interesting path of research.

Acknowledgments. We are very grateful to Lucy Garnier for proofreading this paper and asking many not so naïve questions helping to improve its contents.

References

1. vonAhn L. et Dabbish L. (2004) Labelling Images with a Computer Game. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pp. 319-326.
2. Kipfer B.A. (2001). *Roget's International Thesaurus*, sixth edition, Harper Resource (First Edition : 1852)
3. Landauer, T. K., Foltz, P. W., Laham, D. (1998). An introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.
4. Lapata M. et Keller F. (2005) Web-based Models for Natural Language Processing. In *ACM Transactions on Speech and Language Processing*, vol.2, n°1, pp. 1-30.
5. Lieberman H., Smith D.A. and Teeters A. (2007) Common Consensus: a web-based game for collecting commonsense goals, *International Conference on Intelligent User Interfaces (IUI'07)*, Hawaii, USA
6. Mel'čuk I.A., Clas A., Polguère A. (1995) *Introduction à la lexicologie explicative et combinatoire*, Editions Duculot AUPELF-UREF
7. Miller G.A., Beckwith R., Fellbaum C., Gross D. and Miller K.J. (1990) Introduction to WordNet: an on-line lexical database. In: *International Journal of Lexicography* 3 (4), pp. 235-244.
8. Polguère A. (2006) Structural properties of Lexical Systems: Monolingual and Multilingual Perspectives. *Proceedings of the Workshop on Multilingual Language Resources and Interoperability (COLING/ACL 2006)*, Sydney, pp. 50-59.
9. Robertson S. et Spark Jones K. (1976) Relevance weighting of search terms, *Journal of the American Society for Information Science*, n° 27, pp. 129-146.
10. Salton G. (1968) *Automatic Information Organization and Retrieval*, Mac Graw Hill, NY.
11. Véronis J. (2001) Sense tagging: does it make sense? *Corpus linguistics' 2001 Conference*, Lancaster, U.K.