Blexisma2: a Distributed Agent Framework for Constructing a Semantic Lexical Database based on Conceptual Vectors

Didier Schwab, Lian Tze Lim

Blexisma2: a Distributed Agent Framework for Constructing a Semantic Lexical Database based on Conceptual Vectors

Didier Schwab
Groupe d'Étude en Traduction et en Traitement
des Langues et de la Parole
Laboratoire d'Informatique de Grenoble
Université Pierre Mendès France (Grenoble 2), France
Email: didier.schwab@gmail.com

Lian Tze Lim
Computer-aided Translation Unit
School of Computer Sciences
Universiti Sains Malaysia
11800 Minden, Penang, Malaysia
Email: liantze@gmail.com

***Abstract*—In the framework of meaning representation in Natural Language Processing (NLP), we aim to develop a system that can be used for heterogeneous applications such as Machine Translation, Information Retrieval or Lexical Access. This system is based on six hypotheses which concern meaning representation and acquisition. In this paper, we discuss the related hypotheses that motivate the construction of a such system and how these hypotheses, together with NLP software engineering concerns, led us to conceive a distributed multi-agent system for our goals. We present Blexisma2, a distributed multi-agent system for NLP, its conceptual properties, and an example of inter-agent collaboration. The system is currently being tested on a Grid computing environment.**

***Keywords*—Natural language processing; lexical semantics; multi-agent systems; distributed applications.**

I. Introduction

The representation and use of meaning[1] can help to improve heterougenous NLP applications such as Machine Translation (MT), Automatic Summarisation (AS) or Information Retrieval (IR). In MT, it is well-known that a word can be translated diffently according to the context. For exemple, the English ‹*river*› can be translated in French as ‹*fleuve*› or ‹*rivière*›, the English ‹*mouse*› as ‹*tikus*› (the animal) or ‹*tetikus*› (the computer device) in Malay. In IR, it helps to eliminate documents which contain only inappropriate senses of words in the request, thereby increasing recall and precision. Moreover, an application that "understands" meanings can also help human writers to find better words in composing texts by giving semantically-related words.

We aim to develop a system that can represent and exploit meaning for such applications. For this purpose, we have established six hypotheses based on previous work and experiments. The first two of these concern meaning representation, while and the other four meaning acquisition:

(I) hybrid meaning representation, made up of thematic aspects and lexical aspects;
(II) polysemous nature of lexical items;
(III) automatic generation of acceptions;
(IV) multi-source analysis;
(V) continuous learning; and
(VI) the double-loop process.

We will first present and discuss these hypotheses, how they led us – together with NLP software engineering concerns and heterougenous target applications – to adopt a distributed multi-agent architecture. Conceptual and technical features of its implementation, Blexisma2, will be presented. We conclude with an example of meaning acquisition through collaboration of agents as is now being tested on the USM Campus Grid.

[1]We use *meaning* and *sense* interchangeably in this paper. We do not differentiate between them, nor do we attempt to define either concept.

II. Hypotheses for Constructing a Semantic Lexical Database

The acquisition of lexical meanings is based on six fundamental hypotheses which support the conceptual and implementation architecture chosen for our system.

A. Hypothesis I: Hybrid Meaning Representation

1) The Thematic Aspect – Conceptual Vectors: Vectors have been used in NLP for over 40 years. For information retrieval, the standard vector model (SVM) was invented by Salton [1] during the late 60's, while for meaning representation, latent semantic analysis (LSA) was developed during the late 80's [2]. These approaches are inspired by distributional semantics [3] which hypothesises that a word meaning can be defined by its co-text. For example, the meaning of ‹*milk*› could be described by {‹*cow*›, ‹*cat*›, ‹*white*›, ‹*cheese*›, ‹*mammal*›, ...}. Hence, distributional vector elements correspond directly (for SVM) or indirectly (for LSA) to lexical items from utterances.

The conceptual vector model is different as it is inspired by componential linguistics [4] which holds that the meaning of words can be described with semantic components. These can be considered as atoms of meaning (known as primitives [5]), or also only as constituents of the meaning (known as semes, features [6], concepts, ideas). For example, the meaning of ‹*milk*› could be described by {*LIQUID*, *DAIRY PRODUCT*,

WHITE, *FOOD*, ...}. Conceptual vectors model a formalism for the projection of this notion in a vectorial space. Hence, conceptual vector elements correspond to concepts indirectly, as we will see later.

For textual purposes[2], conceptual vectors can be associated to all levels of a text (word, phrase, sentence, paragraph, whole texts, ...). As they represent ideas, they correspond to the notion of *semantic field*[3] at the lexical level, and to the overall thematic aspects at the level of the entire text.

Conceptual vectors can also be applied to lexical meanings. They have been studied in word sense disambiguation (WSD) using isotopic properties in a text, i.e. redundancy of ideas [6]. The basic idea is to maximise the overlap of shared ideas between senses of lexical items. This can be done by computing the angular distance between two conceptual vectors.

2) The Lexical Aspect – Lexical Relations: We have shown in previous publications [7, 8] that conceptual vectors and lexical relations are mutually complementary, both for semantic analysis and for representing semantic relations between terms (such as synonymy or antonymy). Using lexical relations together with conceptual vectors to represent meaning is a natural consequence of this point.

B. Hypothesis II: Polysemy of Lexical Items

One reason for the difficulty of NLP is that of language ambiguity, including lexical ambiguity, where multiple meanings are associated with the same surface lexical forms (polysemy), e.g. ‹*gravity*›, ‹*pitcher*›, ‹*bank*› and ‹*to take off*›. The relations between the different meanings of a lexical item may be considered as its *internal semantic relations*.

Polysemy must be taken into consideration while constructing the lexical semantic database. We hold that a lexical disambiguation task is impossible without having both thematic (or conceptual) and lexical (or linguistic) information available.

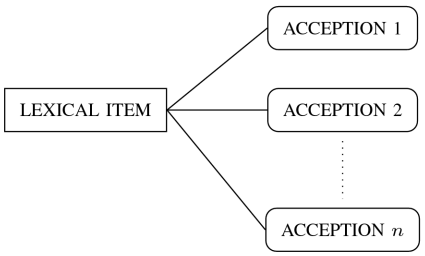
Our database will store objects called *acceptions*. An acception is a particular meaning of a lexical item acknowledged and recognised by common usage. For example, we might consider ‹*mouse*› as having three acceptions: the nouns for the ‹*computer device*› and for the ‹*rodent*›, the verb for the act of ‹*hunt*›ing of the animal (Fig. 1). Contrary to lexical items, acceptions are monosemic (having only one meaning).

Taking into account these first two hypotheses, the acception objects stored in the database contain the following linguistic information:

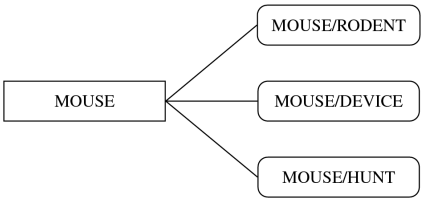
- **morphological** information, including the grammatical category (noun, verb, adjective, etc.), gender (masculine, feminine, neuter) and number (singular, plural),
- number of **hits**, i.e. number of times this acception was retrieved or referred to in corpora,
- **lexical relations**,
- **etymological** information, and
- **gloss**, i.e. brief descriptions to distinguish word meanings,
- **conceptual vectors (CV)** describing thematic information.

For example, for the ‹*rodent*› meaning of ‹*mouse*›, we have:

- **morphology**: noun
- **hits**: 10 000
- **lexical relations**: hypernym = ‹*rodent*›
- **etymology**: Old English : mus
- **gloss**: any of numerous small rodents...
- (main components) of **CV**: {*ANIMAL*, *RODENT*, *LITTLE*}



(a) Overall organisation for sense representation of a lexical item. The conceptual vector of the item is computed from the vector of each of its acceptions.



(b) Overall organisation for sense representation for the lexical item ‹*mouse*›.

Fig. 1. Overall organisation for sense representation of a lexical item.

C. Hypothesis III: Automatic Generation of Acceptions

In our experience on the French language and traditional French dictionaries (e.g. *Larousse* or *Robert*), about 55 % out of over 120 000 entries are polysemous. These have an average of 5 definitions each, thus presenting us with over 450 000 objects (lexical items *and* acceptions) to be processed and stored in the database. Our third hypothesis is to automate this learning task using information from various sources, including dictionaries, synonym and antonym lists, handcrafted indices, web search results, etc. We build and store a *lexie* object for each definition from each source, with the same internal structure as that of an acception (cf. end of section II-B).

Extracting morphological, etymological and gloss information from dictionaries can be automated relatively easily. The conceptual vectors, on the other hand, need more work: it would be unthinkable to manually assign values to each vector element for each lexie. However, the process could be automated with a bootstrapping approach. A subset of acceptions is first selected, based on their frequency of usage in language and/or their polysemy rate. This *kernel* set of acceptions is very much reduced in number, and their conceptual vector elements can be manually chosen and indexed (and thus considered

[2]Conceptual vectors can be associated with any content, not only text: images, videos, multimedia, Web pages, ...

[3]The semantic field is the set of ideas conveyed by a term.

relevant and coherent). The learning process bootstraps from this kernel, building new vectors for new definitions (outside the kernel set) from existing vectors. The underlying principle is that bootstrapped learning from a reduced set of relevant items will ensure coherence between vectors and thus generate a relevant conceptual vector database.

D. Hypothesis IV: Multi-Source Analysis

Definition and gloss texts from dictionaries need to be processed and interpreted with care, especially with respect to metalanguage (the language used to structure the entries), which differ from one dictionary to the next. Phrases like *connected with*, *concerning* and *used to express* are part of the dictionary metalanguage, rather than being actually the definition itself (e.g. *clerical: connected with office work*). Unfortunately, it is sometimes difficult to automatically distinguish between metalanguage and "useful" content in definition texts, as dictionaries seldom indicate the metalanguage vocabulary used. This is why we opt for a multi-source analysis: to statistically temper the various local incoherencies. If a definition from one source is not well-formed (difficult to analyse), another definition from a different source will help mitigate its negative effects.

Another reason is that no one single source can cover the whole vocabulary of a language, not only because language evolves continuously, but also because systematic compilation of all vocabularies in all domains by humans is a huge, difficult task. A multi-source approach would maximise coverage for non-common words or acceptions. For example, the French word ‹*lithurgiste*› can be found in the *Larousse* dictionary [9], but not in *Robert* [10].

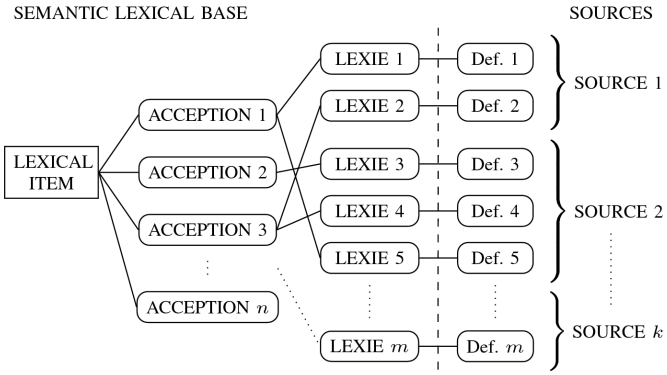


Fig. 2. Overall organisation of the meaning representation for a lexical item. Lexies are built from definitions from different sources. These lexies are then grouped or aligned to build acceptions corresponding to each sense of the term.

Fig. 2 shows the overall organisation of the semantic lexical database. We obtain definitions for a lexical item from a number of sources, building a lexie (made up of lexical information and a conceptual vector) for each definition. These lexies are then categorised to build acceptions for the lexical item.

For example, the following definitions for the lexical item ‹*mouse*› are found from three online dictionary sources[4], namely *MSN Encarta* (MSN) [11], the *Merriam-Webster Online Dictionary* (MW)[12] and the *Cambridge Advanced Learner's Dictionary*[13]:

mouse.1 : #noun# any of numerous small rodents (as of the genus Mus) with pointed snout, rather small ears, elongated body, and slender tail. *[MW]*
mouse.2 : #noun# a small mobile manual device that controls movement of the cursor and selection of functions on a computer display. *[MW]*
mouse.3 : #verb# to hunt for mice. *[MW]*
mouse.4 : common name for any small member of three families of rodents; large species of one of the families to which mice belong are known as rats... *[MSN]*
mouse.5 : Mouse (computer), a common pointing device, popularized by its inclusion as standard equipment with the Apple Macintosh... *[MSN]*
mouse.6 : #noun# a small mammal with short fur, a pointed face, and a long tail. *[Camb]*
mouse.7 : #noun# a small device which you move across a surface in order to move a pointer on your computer screen. *[Camb]*

From this list, definitions $\{1, 4, 6\}$ can be grouped into one acception (meaning) relating to a *rodent*; definitions $\{2, 5, 7\}$ into another for a *computer device*, and $\{3\}$ into a third for *hunt*. Fig. 3 then shows the overall sense organisation for ‹*mouse*› from these seven definitions.

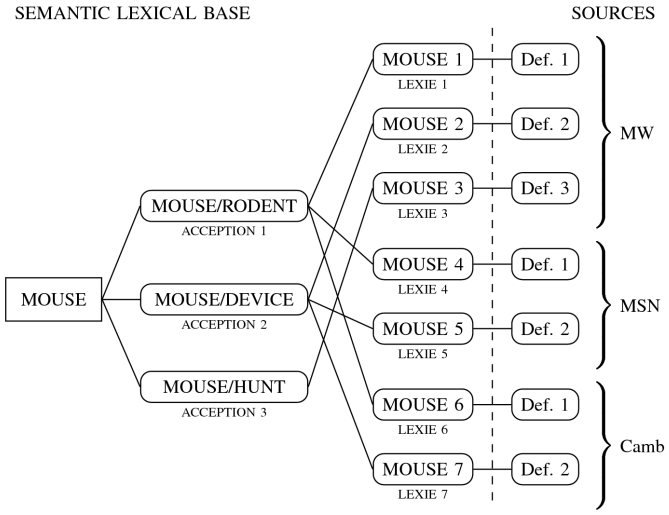


Fig. 3. Overall organisation of meaning for the lexical item ‹*mouse*›

E. Hypothesis V: Continuous Learning

Certain types of corpora (especially newspaper articles) require named entity detection (e.g. personalities, organisations) and neologism[5] detection for coherent interpretation, as well as providing clues about the domain. For example, the occurrence of the term ‹*Arcelor*› indicates a high probability that the document is related to steel manufacturing. Such information

[4]Note that the definitions given here might not be the most current version at the original websites

[5]Newly coined terms, usually for a specific professional domain.

and appropriate knowledge structure may be acquired by learning from online news servers.

Furthermore, it would be imprudent to assume that the constructed conceptual vectors would be coherent after only a single learning cycle. Vector convergence towards a quasi-stable condition can only be achieved after a large number of cycles, as "key" words in definitions will only become prominent after many iterations. The exact number is difficult to estimate accurately beforehand, but one would expect it to be related to the size and richness of the vocabulary used in the definitions. Such lexical variability, and the inherent impossibility to completely "freeze" the database of vectors, lead to another hypothesis: that the semantic lexical database needs to undergo iterative updates via continuous learning.

F. Hypothesis VI: The Double Loop

The double loop [14, 15] is an invariant structural element which allows action on its environment and is itself a product of this action. In a human individual, this structure exists at all levels, from the lowest (that of a cell) to the highest (that of the entire body, including the central nervous system). This organisation of flows of actions in a loop allows modification of the behaviour of the structure *inside* the environment, while in the opposite direction, changes are also propagated to the elements *of* the environment. In other words, regular exchanges between the structure and its environment cause changes to both parties. A behaviour is reflected in regular flows to stabilise the structure that initiated it. On the contrary, maladaptive behaviours result in irregular flows, leading to the gradual destruction of the original structure.

We seek to adopt this mutual regular exchange, not between the structure and environment of the human body, but between those of language usage. It has been shown previously [7, 16–19] that a conceptual vector can be improved by the effects of lexical functions, and also that the lexical function data themselves are significantly enhanced by the use of lexical information and of the corresponding vectors. Hence, the two processes of learning of lexical functions and the construction of new vectors are mutually benefitting each other, each feeding their outputs into the inputs of the other. This mutually enriching phenomena bears a certain resemblance to the principle of the double loop, which we adopt as our sixth hypothesis.

These six hypotheses, together with the requirement of robustness, software engineering, and the distribution of large amounts of data across multiple machines, have led us to choose a distributed multi-agent architecture for our system, called Blexisma2, as will be discussed in the next section.

III. Towards a Society of Agents

As mentioned previously, our goal is to implement a system that allows the automatic learning of semantic information (encompassing thematic aspects in the form of conceptual vectors, and lexical aspects in the form of semantic relations) and its retrieval for use. The tasks involved include fetching new definitions and analysing them with already computed lexies and acceptions, either to build new objects or to revise existing ones (cf. section II). Analysing definitions is not always sufficient to ensure coherence of the vectors, as there may be problems due to metalanguage or other factors. Complementary solutions are then possible, for instance by using semantic relations between lexical items (synonymy [20], antonymy [17], hypernymy, ...). These relations are useful at two levels: during construction of acceptions, and during revision of existing acceptions. On the other hand, there are many different applications for the semantic lexical database: for lexical disambiguation, annotation, lexical transfer, information retrieval and more. With so many demands on both the learning and the application front, it is necessary to find a mechanism for easily adding and extending functionalities. This is the main reason that our main architecture took the form of a multi-agent system (MAS).

A. MAS and NLP

Multi-agent systems originated from the field of distributed artificial intelligence (DAI), which "share intelligence among agents". This intelligence is a consequence of the agent interactions, i.e. by emergence. An agent is a physical or virtual entity, able to have perception of and act on its environment, to communicate directly with other agents, to own resources, has some kind of competencies, and to offer services [21]. MAS have been used before in NLP. Research in DAI on speech comprehension (HEARSAY-II [22]) had been carried out in the early 1970's. More recently, Lebarbé [23] used MAS for syntactic analysis, and Menézo et al. [24] for error detection. Other research, including CARAMEL [25] and TALISMAN [26], are similar to our own in their aims, i.e. modular architectures for semantic information learning, retrieval and application.

B. Why Choose a Distributed MAS?

This section further discusses the reasons for choosing a distributed architecture for our system. The rationales are not only limited to our hypotheses for building a semantic lexical database (section II), but also take into account technical considerations.

1) Implications of Hypotheses: While the first two hypotheses (those concerning hybrid meaning representation and the joint use of lexical items and acceptions) are easily compatible with any NLP tool whatever its architecture, the remaining hypotheses on automatic generation, multi-source analysis, continuous learning and double loop all indicate that a multi-agent approach is desirable. In particular, since we would like to acquire as much information and thematic vocabulary as possible from an ever expanding list of sources (dictionaries, Web content, various vocabulary lists), it would seem that the use of independent agents would be the most straightforward solution.

On the other hand, a multi-agent architecture would also be well-equipped to address the requirement for "self-improvement" with a double loop (section II-F). Each agent modifies its knowledge base of conceptual vectors using the

lexical information encountered or deduced. Other agents constituting its environment would also benefit from its improvements, which would be fed into their own processes (see section IV-A1b). Hence, the application and learning of conceptual vectors are closely linked.

2) Implications of Target Applications: Eventually, we aim to develop a variety of heterogenous applications, which will draw on the semantic lexical database. These applications would cover areas such as machine translation, automatic summarisation, information retrieval, knowledge extraction and others, and would require software components built around the database. These software components, each with their own specific competencies, should be easily integrable into new NLP applications, and even for multiple languages. To illustrate, picture agents specialising in translation, information retrieval, or summarisation respectively, each providing a particular service and inter-communicable with each other, working together in some large "intelligent" document analysis application. It is with this view in mind that we adopt a MAS architecture.

3) Technical Reasons:

a) Distribution across Multiple Machines: NLP systems are well-known to consume high system resources, due to the significant amount of data to be stored (e.g. a typical French lexicon would contain at least 100 000 entries, not yet including proper names) and the heavy computations. In our case, each agent must maintain its own knowledge base in its own memory, the size of which largely depends on the length of the conceptual vectors. Our (still small at time of writing) database for the English language has around 1.2 GB worth of data on disk, and will increase in future. It would then be difficult to maintain tens or hundreds of agents on the same machine, Moore's Law notwithstanding. A cheaper solution would be to host agents on different, less powerful machines in a distributed environment, so that each agent can optimise the available resources.

b) Software Engineering: Modularity is an important characteristic of agent architectures, since it allows easy deployment of additional modules. Many legacy NLP tools, developed in obsolete programming languages or for specific types of computer architectures, are still currently in use. Rebuilding these legacy tools is not always feasible, and yet it would usually be difficult to integrate them into our own applications. However, using an agent architecture, it would be easy to create agents that acts as interfaces between these legacy applications and our own system. For instance, we may create a morpho-syntactic analyser agent by interfacing to Connexor's *Machinese Syntax* parser[27]. The modularity afforded by agents also facilitates combination of several systems into a single one.

By using a multi-agent system, a new agent with new or improved functionality can be launched at any time without having to halt and restart the entire system. This means the learning process can be continued without having to wait for new heuristics or additional functionalities to be implemented, tested and upgraded, thus saving time. Moreover, restarting the entire system would mean that all data would have to be reloaded during initialisation, which will certainly take some time (depending on the system specifications of the machine).

IV. Blexisma2

A. Conceptual Features

We now present, on a conceptual level, the internal organisation of our agents, as well as the overall organisation of our system, Blexisma2. Note that the features discussed here are not tied to any underlying implementation techniques.

1) Agents: We examine the characteristics that typifies the cognitive aspects of our agents, i.e. how they reason. We also describe their social organisation in three levels (agent, role and language) at which they interact and communicates.

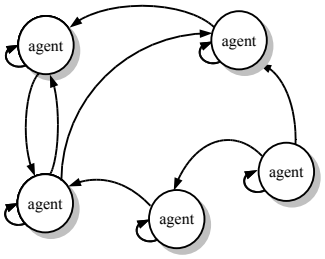
a) Recursive Agents: Like most MAS, our system can be viewed at different scales. Each agent may itself be composed of reactive agents whose emerging effects will be forwarded to other agents by sending messages. For example, to analyse a text, an agent in a system running an ant algorithm may disambiguate the lexical items on the leaves of a corresponding morpho-syntactic tree, and propagate the conceptual vector of the overall text to the main agent of the application [7].

b) Re-inforced Learning in a Double Loop: Each agent maintains its own knowledge base, initialised by loading from the semantic lexical database and continuously modified according to its experience and interactions with other agents. They take advantage of the supplied information to update their knowledge before responding to a query. For example, a learning agent might extract a list of antonyms for a particular lexie, as in the following:

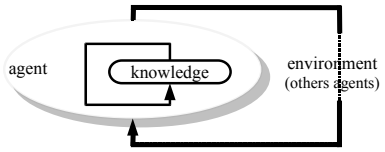
to borrow.1: get temporarily. *Antonyms: to lend, to loan.*

The learning agent might then request an antonymy agent to compute and provide the corresponding vectors. The antonymy agent would first use the information provided by the learner agent to update its own store of vectors (internal loop). After completing the learner agent's request, it sends back the computed vectors, which would be added to the entire system's collection of vectors (external loop). These agents therefore co-operate to improve the coherence of the database of vectors. On the other hand, by using information received from the same database, these agents will also adapt their computations accordingly to deliver better results [16]. In this way, the overall system and the individual agents mutually enrich and benefit each other (cf. Fig. 4 and section II-F).

c) Competition Breeds Excellence: When an agent may request other agents to help when faced with a task. These other agents might provide different solutions, based on different approaches that each are implemented by. For example, if a semantic analysis agent encounters a possible pattern, it might ask agents specialising in antonymy, hypernymy or meronymy if they can characterise that particular pattern. It might even ask other semantic analysis agents for their opinions. In both cases, the initiating agent will then consider all responses received, assigning weights according to majority votes or on confidence scores supplied by the respondent agents. In this



(a) Macroscopic organisation of a system and the interaction of the different agents.



(b) Microscopic organisation of a system, the "world vision" of an agent. External data is used to improve the agent's knowledge database; the agent in turn improves external data (double loop).

Fig. 4. The Double Loop in Blexisma2

way, the effects of noisy or false results from some agents may be cancelled out by the more reliable results.

d) Uniqueness of Agents: Each agent is unique during its lifetime, despite the fact that multiple agents may have the same role or specialises in the same area (for instance, there may be several agents responsible for carrying out analysis on French documents, or specialising in the same semantic relationship). These agents might tackle the same problem with different approaches or algorithms. They might also be simply copies of each other in that they are of the same source code. In the latter case, however, each agent is still unique, due to its "experience" gained through interactions and data exchanges with other agents. For example, a learning agent would select random objects in its knowledge base to be periodically updated, rather than imposing some particular order to the selection, since it would be hard to determine if there exists any particular order that would guarantee faster convergence anyway. Each learning agent would acquire a unique experience as it analyses its objects in a different order. The same thing would happen with semantic relation agents since they may not encounter the same pairs of terms, or at the same frequencies.

2) Social Organisation: All Blexisma2 agents have the following three attributes:

- name – each agent is known by a name which is its unique identifier.
- language – the "mother tongue" of the agent, the natural language which it works with. An agent may be independent of any language if it does not use any lexical information.
- role – the function played by the agent in the system, e.g. basic learning analyser, morpho-syntactic analyser, experts in various lexical functions and semantic relations, contextualiser, etc. The role attribute does not presume anything about how an agent carries out an action, merely what actions it is capable of.

It should be noted that this organisation of {*Agent, Language, Role*} is quite similar to Gutknecht and Ferber's [28] model of {*Agent, Group, Role*}, with the difference that we do not anticipate that an agent in Blexisma2 would have multiple roles or belong to multiple groups (i.e. languages).

3) Communication via Message Sending: The agents communicate with each other by sending messages, in one of two possible ways:

- *Direct communication between agents:* An agent sends a message directly to another agent. The recipient agent may perform some action, based on its interpretation of the message, and respond accordingly or indicate its incompetence to respond satisfactorily.
- *Broadcasting to all agents:* An agent may send a message to all agents with a given role, all agents of a given language, all agents of a given role of a given language, or simply all agents. Each recipient agent may perform some action in response, and possibly send back a reply message to the issuing agent.

At a more local scope, the agents responsible for interfacing with the lexical semantic database (and therefore plays the part of a centralised lexical information broker) can also be considered as a "blackboard", on which all learning agents records the updates calculated, together with the time of modification. All these information are visible to all agents.

B. Implementation features

We have built a prototype system for Blexisma2 following the principles advocated above using *MadKit*[29], a modular and scalable multi-agent platform written in Java. The agents themselves may be programmed in languages other than Java. Each agent has specific competencies, and interacts with other agents by asking for services or responding to such requests. Agent management is handled by the *MadKit* kernel. When each agent is created ("goes live"), it informs the kernel of its identifier, role and language (English, French, etc.), as well as other technical information such as the host name and port number it is running on. The kernel accepts and admits this new agent if the identifier is unique. During the lifetime of the agents, when one agent intends to address a request or message to another agent, the *MadKit* kernel would dispatch the message to competent agents according to their roles and availability.

We have currently implemented the following Blexisma2 agents:

- Lexicon Dispensers – responsible for interfacing with the on-disk database management system storing all data, to distribute information from the semantic lexical database (as explained in II-D) to other agents that require them.
- WordNet Logical Function Analysers – these are actually agents of the *Conceptual Vector Learner* family, which uses pre-parsed and sense-tagged gloss text from *eXtended WordNet* [30, 31] to generate and update conceptual vectors for English lexies found in WordNet [32].

- k-NN Seekers – retrieves the k-th semantically closest lexies or acceptions to any given lexie (k-nearest neighbour search). For example, given ‹*computer*›, such an agent might return {‹*keyboard*›, ‹*website*›, ‹*software*›, ...}.

These agents are now under testing on a Grid computing environment. We are also planning other families of agents in the future:

- Lexical Function Extractors – extracts valid lexical function mappings between items from specialised dictionaries, from definitions in traditional dictionaries or from web corpora.
- Lexical Functions Experts – responsible for computing new conceptual vectors from the respective semantic relations that each agent specialise in, like synonymy [20], antonymy [16], hypernymy [33] or other lexical functions.
- Categorisers – groups and categorises lexies into acception objects.
- Validators and Rectifiers – checks and validates the coherence of data in the semantic lexical database and possibly rectifies it.

C. Example of Interaction Between Agents

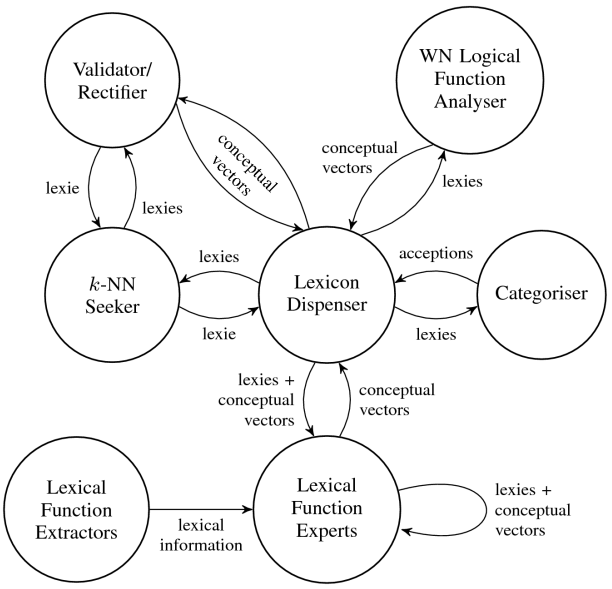


Fig. 5. Macroscopic organisation of the system during a semantic analysis.

We now describe an example scenario of inter-agent collaboration during the construction of a conceptual vector (Fig. 5). The WordNet Logical Function Analyser asks the Lexicon Dispenser for a lexie from *eXtended WordNet*, including the pre-parsed and sense-tagged gloss text. The Analyser further asks the Lexicon Dispenser for the list of relevant lexies as indicated by information in the initial lexie. It uses the conceptual vector of these lexies to compute the new vector of the initial lexie. If any of the lexies in the reference list does not yet have a vector, the Analyser might recursively compute one for it, or randomly generate one (see [8] for the full algorithm). On the other hand, the Lexical Function Extractor would retrieve whatever lexical function pairings it may find from various sources, and pass the term pairs to the respective Lexical Function Experts. These experts then retrieves existing conceptual vectors from the Lexicon Dispenser, and generate new vectors for the semantic lexical database using information from both the Dispenser and the Extractors. Different types of Lexical Function Experts might "discuss" among themselves before producing their respective results. Validators and Rectifiers would communicate with the k-NN Seeker to double check that the "neighbourhood" of each lexie is reasonably relevant (e.g. ‹*restaurant*› would be within a narrow neighbourhood of ‹*chef*› but probably not ‹*police*›), and applies remedial actions when they encounter discrepancies. In parallel to all this action, Categorisers would also be going through the lexies via the Lexicon Dispenser to group lexies into acceptions.

V. Conclusions

We have presented the architecture of a generic semantic lexical database useful for NLP applications requiring semantic analysis. We discussed the six hypotheses taken to design this database, namely the *hybrid meaning representation* of both thematic and lexical aspects; *polysemous* nature of language; and the *automation* of *continuous* learning tasks from *multiple sources* in a *double loop*. Finally, we demonstrated the architecture of a prototype distributed MAS system, Blexisma2, by describing the characteristics and working of the agents in the system.

We will continue to work on each agent to enhance their performance, and to deploy them in a Grid environment. New agents will be added to implement new functionalities or different heuristics. We will also work to improve the existing agent communication channels to better utilise the capabilities of the Grid.

Acknowledgement

The work reported in this paper was carried out as part of a Universiti Sains Malaysia Research University Grant project entitled "Grid Computing Cluster – The Development and Integration of Grid Services & Applications". We thank Dr Chan Huah Yong and his great team of researchers at the Grid Computing Laboratory, Universiti Sains Malaysia, for their permission and help in using their Grid facilities.

REFERENCES

[1] G. Salton, "The Smart document retrieval project," in *Proc. of the 14th Annual Int'l ACM/SIGIR Conf. on Research and Development in Information Retrieval*, Chicago, 1991.

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, 1990. [Online]. Available: http://citeseer.nj.nec.com/deerwester90indexing.html

[3] Z. S. Harris, M. Gottfried, T. Ryckman, P. Mattick Jr., A. Daladier, T. Harris, and S. Harris, *The form of Information in Science, Analysis of Immunology Sublanguage*, ser. Boston Studies in the Philosophy of Science. Kluwer Academic Publisher, Dordrecht, 1989, vol. 104.

[4] L. Hjelmlev, *Prolégolème à une théorie du langage*. éditions de minuit, 1968.

[5] A. Wierzbicka, *Semantics: Primes and Universals*. Oxford University Press, 1996.

[6] A. J. Greimas, *Structural Semantics: An Attempt at a Method*. University of Nebraska Press, 1984.

[7] D. Schwab and M. Lafourcade, "Lexical functions for ants based semantic analysis," in *ICAI'07- The 2007 International Conference on Artificial Intelligence*, Monte Carlo Resort, Las Vegas, Nevada, USA, June 2007.

[8] D. Schwab, L. T. Lim, and M. Lafourcade, "Conceptual vectors, a complementary tool to lexical networks," in *Proc. of the 4th Int'l Workshop on Natural Language Processing and Cognitive Science (NLPCS 2007)*, Funchal, Madeira, Portugal, 2007.

[9] Larousse, Ed., *Le Petit Larousse Illustré 2001*. Larousse, 2001.

[10] L. Robert, Ed., *Le Nouveau Petit Robert, dictionnaire alphabétique et analogique de la langue française*. Éditions Le Robert, 2000.

[11] "http://encarta.msn.com."

[12] "http://www.merriam-webster.com."

[13] "http://dictionary.cambridge.org."

[14] C. Lecerf, *Une leçon de piano ou la double boucle de l'apprentissage cognitif.* Université Paris 8, Vincennes Saint-denis: Université Paris 8 - Vincenne-Saint-Denis, Mars 1997, vol. 3-1997, revue Travaux et Documents.

[15] ——, "Tackling the stability/plasticity dilemma with double loop dynamic systems," in *ESANN'1999 proceedings - European Symposium on Artificial Neural Networks*, April 1999, pp. 153–158.

[16] D. Schwab, M. Lafourcade, and V. Prince, "Antonymy and conceptual vectors," in *COLING'2002 : 19th International Conference on Computational Linguistics*, vol. 2/2, Taipei, Taiwan, August 2002.

[17] ——, "Vers l'apprentissage automatique, pour et par les vecteurs conceptuels, de fonctions lexicales. l'exemple de l'antonymie," in *TALN 2002*, vol. 1, Nancy, June 2002.

[18] ——, "Extraction semi-supervisée de couples d'antonymes grâce à leur morphologie," in *TALN'2005*, Dourdan, June 2005.

[19] D. Schwab and M. Lafourcade, "Modelling, detection and exploitation of lexical functions for analysis," *ECTI Journal*, vol. 2, no. ISSN 1905-050X, pp. 97–108, 2007.

[20] M. Lafourcade and V. Prince, "Synonymies et vecteurs conceptuels," in *TALN'2001*, Tours, France, July 2001.

[21] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Paperback, 1999.

[22] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The HEARSAY-II speech-understanding system: Integrating knowledge to resolve uncertainty," *ACM Comput. Surv.*, vol. 12, no. 2, pp. 213–253, 1980.

[23] T. Lebarbé, "Vers une plate-forme multi-agents pour l'exploration et le traitement linguistique," in *TALN'2001*, Tours, France, July 2001.

[24] J. Menézo, D. Genthial, and J. Courtin, "Reconnaissances pluri-lexicales dans celine, un système multi-agents de detection et correction des erreurs," in *NLP+IA96 : International Conference on Natural Language Processing and Industrial Applications.*, 1996.

[25] G. Sabah, "CARAMEL: Un système multi-expert pour le traitement automatique des langues." *Modèles linguistiques*, vol. tome XII, fascicule 1, 1990.

[26] M.-H. Stéfanini and K. Warren, "A distributed architecture for text analysis in french : an application to complex linguistic phenomena processing," in *COLING'1996 : 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, 1996.

[27] "http://www.connexor.eu/technology/machinese/machinesesyntax/."

[28] O. Gutknecht and J. Ferber, "Vers une méthodologie organisationnelle de conception de systèmes multi-agents," in *JFIADSMA'99 : Actes des 7èmes Journées Francophones d'Intelligence Artificielle et Systèmes Multi-Agents*, 1999.

[29] "http://www.madkit.org/."

[30] S. M. Harabagiu, G. A. Miller, and D. I. Moldovan, "Wordnet 2 - a morphologically and semantically enhanced resource," in *Workshop SIGLEX'99 : Standardizing Lexical Resources*, 1999.

[31] R. Mihalcea and D. Moldovan, "eXtended WordNet: Progress report," in *Proceedings of NAACL 2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, USA, 2001.

[32] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An on-line lexical database," *International Journal of Lexicography (special issue)*, vol. 3, no. 4, 1990.

[33] D. Schwab and M. Lafourcade, "Hardening of acception links through vectorized lexical functions," in *PAPILLON'2002*, Tokyo, Japon, Août 2002.