Language Games: Learning Shared Concepts Among Distributed Information Agents

Conor Hayes and **Paolo Avesani** and **Marco Cova**
ITC-IRST,
Via Sommarive 18 - Loc. Pantè, I-38050 Povo, Trento, Italy
hayes, avesani, cova@itc.it

Abstract

Early agent research recognised that co-operating agents require access to unambiguous, semantic description of the same concept, entity or object. In fact, agent-based research on this problem anticipates many of the current initiatives of the Semantic Web project. The proposed solution involves developing a domain-specific ontology that can be mapped to other ontologies as required. In this paper we describe an alternative approach which allows autonomous agents to index shared objects without requiring ex-ante agreement on an ontology. Using a process of distributed negotiation, each agent builds a lexicon of the problem-solving competences of other agents. We present an overview of our work using this approach in three domains: a web services scenario, a multi-case-based agent approach and finally, Tagsocratic, a blog-indexing service. We then describe our future work on several open issues related to this research.

1 Introduction

The problem of interoperability between autonomous, non-centralised software components has been an intrinsic feature of agent research. Although agent communication languages (ACLs) such as KQML [Finin *et al.*, 1994] and FIPA [FIPA, 2000] provide standard communication protocols, there is still the problem of agreement on a common language with which to allow agents to co-operate about content in their environment. One of the key incentives of the Semantic Web project was to introduce a semantic framework that would enable the processing of Internet resources by intelligent entities such as software agents. In pursuit of this, RDF and related initiatives such as DAML and OIL allow domain-specific ontologies to be specified [McGuinness *et al.*, 2002]. The usual strategy of these efforts consists in establishing a relationship between the local representations and a common reference encoding, namely a shared ontology. This approach requires two steps:

1. The definition of an ontology for the specific domain.
2. The definition of a mapping between a local representation and the shared ontology.

While intuitive, these approaches are often not effective in practice. The first step requires ex-ante agreement between all potential users of the ontology. Furthermore, the mapping step is generally far from trivial and often requires manual intervention. Indeed, in terms of take-up on the WWW, the top-down proposals of the Semantic Web have been less successful than simpler bottom-up protocols such as RSS,[1] which enable information providers to publish information in standard form quickly without having to agree on semantics in advance. The latter scenario, however, still implies ex-post agreement on semantics between locally defined representations. In terms of agent technology, this means allowing the agent complete autonomy in representing the information objects it expresses, but also requiring it to learn a mapping between alternative representations held by other agents.

In contrast, we present an alternative perspective in which autonomous, distributed agents negotiate using a technique called *language games* in order to develop a distributed indexing lexicon [Steels and McIntyre, 1999]. By learning this lexicon each agent builds a picture of the competences of external agents and can quickly request resources from these when it is not able to solve a problem locally. To illustrate the technique we present three scenarios in the area of distributed information retrieval. In the first, a web services plug-in allows distributed information providers to learn which topics they have in common. In the second, distributed case-based information agents learn the competences of each other so that information that is not available locally may be retrieved from other competent agents. Finally, we introduce Tagsocratic, a project in the blogging domain in which agents communicate in order to learn topic alignments between bloggers. In this scenario, the blogger is enabled to quickly find posts by other bloggers on the same or similar subjects.

In section 2, we locate this work in relation to the Semantic Web project and to previous work in agent research. Sections 3 and 4 describe the language games technique. We characterise this technique as having a two stages: an eager indexing stage and a problem-solving stage. In section 5, we describe our initial work in applying this technique in three domains. We describe open issues and future work in section 6.

[1] blogs.law.harvard.edu/tech/rss

2 Background

Early agent research recognised that co-operating agents require access to unambiguous semantic description of the same concept, entity or object. For example, the DARPA Knowledge Sharing Effort (KSE) tackled the problem of interoperability between heterogeneous knowledge sources by dividing the problem into three layers: a language translation layer, a communications layer and an ontology layer in which semantic consistency is preserved within the domain in which the applications work [Neches *et al.*, 1991]. The key idea is that a shared domain-specific ontology is developed in advance from which agents can choose or extend the elements that best suit their own perspective. Knowledge can be shared because agents can translate between their own local representation and the shared ontology. Agent communication languages such as KQML and FIPA assume a shared ontology between communicating agents. In KQML, the message layer allows the agent to specify the ontology associated with the message, while FIPA has specified an ontology service to allow agents to reason about domain knowledge.

Indeed, one of the key objectives of the Semantic Web project is to enable processing of web resources by distributed, intelligent entities such as software agents [Berners-Lee *et al.*, 2001]. As such, this project has produced several specifications such as RDF, OIL and DAML, which allow domain-specific ontologies to be produced. The Semantic web proposals can be viewed as a top-down approach to the problem of semantic agreement: agreement is reached in advance on the formal relations between entities in a particular domain, after which agents or other intelligent software applications can reason about the objects in the domain. However, a difficulty in this approach is how agreement is reached on the correct knowledge representation for a particular domain. If two or more ontologies are used for a particular domain, agents who wish to communicate will require a translation service between ontologies. In the worst case scenario, where agreement is not reached, each agent uses a knowledge representation based on local semantics and communication between agents requires a translation service between each pair of agents.

Despite the obvious benefits of an agreed semantic framework, the take-up on Semantic Web proposals to date has been slow. Instead, simpler, 'bottom-up' initiatives have become much more successful. For example, RSS has become the standard means of allowing information providers to publish up-to-date data on the web without requiring explicit semantic mark-up. In the blogosphere, blog software enables bloggers to mark-up each of their posts with locally defined categories. Two key observations can be made here: the proliferation of these 'bottom-up' approaches appears to be stimulated by the lack of centralised co-ordination required for their deployment. This would seem to suggest that web content providers prefer minimal constraints on the local definition of semantics. Secondly, the issue of semantic alignment appears to be addressed by a second wave of low level initiatives such as RSS aggregators and category aggregators [2].

[2] http://www.technorati.com/

There are also a number of approaches to matching heterogeneous schemas based on machine learning [Rahm and Bernstein, 2001]. However, where agents have heterogeneous knowledge representations, such approaches require a mapping between representations for every pair of agents. In a similar spirit to this work, [Reed *et al.*, 2002] argue that the specification of the agent communication language (ACL) can be a run-time process where agents tailor their communication primitives to the circumstances in which they find themselves. In contrast, we are concerned with how distributed information agents can learn to refer to common objects in their environment without having to formally define or learn a particular semantic framework.

3 Introduction to Language Games

Our research is based on experiments on language evolution called *language games* [Steels and McIntyre, 1999]. Rather than requiring pairwise translation between heterogeneous knowledge representations, agents achieve a common language through a process of exchanging instances and feedback. The language games technique was developed originally to explore how autonomous, heterogeneous physical agents could develop a common language in order to communicate about objects in their environment. Agents are not constrained in terms of their internal representation about the objects. Instead, they learn a distributed common lexicon which allows them to refer to common objects in their environment although each agent has a different internal representation.

This technique has more recently been adapted to tackle the problem of aligning heterogeneous knowledge representations among information providers on the web [Avesani and Agostini, 2003; Avesani and Cova, 2005]. The problem scenario can be described as follows: The web is a distributed repository of web pages to which autonomous agents (users) add information in a random fashion. Although, many web pages describe related subject matter, the web is not organised in a topic-centric manner. We describe how distributed agents co-operate by game playing in order to produce a distributed index of topics that allows an agent to link its own topic definitions to those defined by agents with similar interests.

Our definition of an agent is quite varied. In the Moleskiing application, an agent is a server side add-on that learns the correspondence between ski-mountaineering trips being offered by other autonomously managed web sites [Avesani and Cova, 2005]. In the CBR agents initiative [Avesani *et al.*, 2005b], each agent filters and collects sites offering travel itineraries for a particular user interest group. A user can query the agent's case base memory for a suitable trip description. If an adequate solution is not found, the agent contacts other agents with competence in the area. The key point is that each agent learns and indexes the competences of external agents using the language games technique. Finally, the Tagsocratic project examines the role for information agents in the Blogosphere [Avesani *et al.*, 2005a]. In this scenario, each blogger has an agent who learns which posts by other bloggers address similar topics to those of its blog master.

4 Indexing and Problem Solving

In this section we give a top level view of the language games methodology. For a more formal introduction to our methodology see [Avesani and Agostini, 2003]. The language games methodology can be viewed as being made up of 2 stages:

1. Indexing phase: An eager index learning stage in which agents communicate to assess similarity between their object representations. During this phase a shared distributed index emerges.
2. problem-solving phase: In this stage each agent can quickly retrieve relevant cases by consulting its indexing lexicon and issuing a label identifier to other agents.

4.1 Indexing phase

The indexing phase consists of a number of language games whereby a community of agents converge on a common set of labels for the objects in their world. The common set of labels constitutes a distributed global lookup table which is then used to reference similar cases in external case bases. For example, if we consider the CBR example previously mentioned and described in more detail later, the index learning phase involves each agent learning the correspondence between similar cases in other agents and assigning those cases a global identifying label by which those cases can be identified. The set of global labels constitutes an index of the global competence of the agent community. However, this index is not maintained centrally but is distributed among the agent community. After the indexing process, an agent can request similar cases to one of its own cases from another agent by simply sending it the appropriate global label. The receiving agent then looks up which case(s) in its case base corresponds to this label and returns it to the requesting agent.

Each language game involves an exchange between a pair of agents where one acts as a speaker and the other acts as a listener. As shown in Table 1, all agents maintain a table which records the degree of association between a global label ($\mathcal{L}_p$) and one of its cases ($\mathcal{O}_p$). In the table, *u* refers to the number of times the label has been used in different language games by this agent while *a* refers to how often it has been successfully used. Each game consists of a communications phase and an assessment phase (see Figure 1).

Communications

The *communications* phase proceeds as follows: The speaker agent chooses one of its object (case) representations and the corresponding best scoring label from the association table. In the example in Table 1, the label l_1 is the best performing label for case c_{j9}, having been successfully used for 8 out of 10 language games. Agent CB_j and agent CB_i agree to have a language game. Agent CB_j, acting as the speaker, selects case c_{j9} and (encodes it as) label l_1. It sends the label to Agent CB_i. CB_i decodes label l_1 using its association table. If it finds that the label corresponds to one of its own cases, it selects the case and returns it to agent CB_j.

Assessment

The next phase of the game is the *assessment phase*. Agent CB_j must assess whether the case sent by agent CB_i is equivalent to case c_{j9}. In the Moleskiing application and the CBR agents application we successfully use a bi-partite string matching algorithm to determine equivalence [Kuhn, 1955]. The cases in the speaker's case base are ranked by similarity to the case received from the listener. If the top ranked case is the same as the case initially chosen by the speaker, the game is deemed to be a success and the speaker accordingly updates its table, increasing the fraction of times the label was successfully deployed (9 times out of an 11). The speaker sends positive feedback to the listener so that it too can update its table. If the game fails - the listener may return nothing or may return a mismatching case, the fraction of successes recorded by the label is reduced to 8 out of an 11. The speaker sends negative feedback to the listener and likewise it reduces the fraction of successful deployments for this label in its table.

$\mathcal{O}_p$	$\mathcal{L}_p$	***u***	***a***
c_{j9}	l_1	10	8
c_{j9}	l_2	3	0
c_{j5}	l_3	5	4
c_{j6}	l_4	8	1

Table 1: The lexicon of case base CB_j during the learning phase

It is important to remember that the assessment phase is domain dependent. For the Tagsocratic project, for instance, we use a naive Bayes classifier [Lewis, 1998] to determine whether the posts received from the listening blog agent are of the same class as the blog posts represented by the global label on the speaking blog agent. In this sense, the assessment phase in Tagsocratic is more difficult in that we have to assess whether exchanged posts belong to the same concept shared by both peers, rather than determining equivalence between object representations.

4.2 Problem-solving Stage

The problem-solving stage allows distributed CBR agents to quickly retrieve remote cases by issuing the learned index label. As similarity computation has been eagerly computed and indexed during the indexing stage, the bandwidth and computation overheads involved in sending and decoding a label are very low, entailing fast query time response.

Let us consider an example scenario in which CBR agents retrieve travel information based on input from a user. Each agent operates autonomously and has a case base made up of a set of travel itineraries. The solution part of each case description consists of reviews and comments posted by other users on the quality of the proposed travel package. Each case base is represented according to a locally defined schema. CBR agents may cooperate so that if a locally retrieved solution is not adequate it can contact agents with competence in the problem area to retrieve alternative solutions. In Figure 2 we illustrate the cycle just described. Each case base agent contains reviews that are pertinent to a particular interest group. While not delivering results to its user base, the agent is busy crawling the web for case material relevant to the interests of its user group. Let us consider the scenario where, after querying case base agent CB_i, the user provides

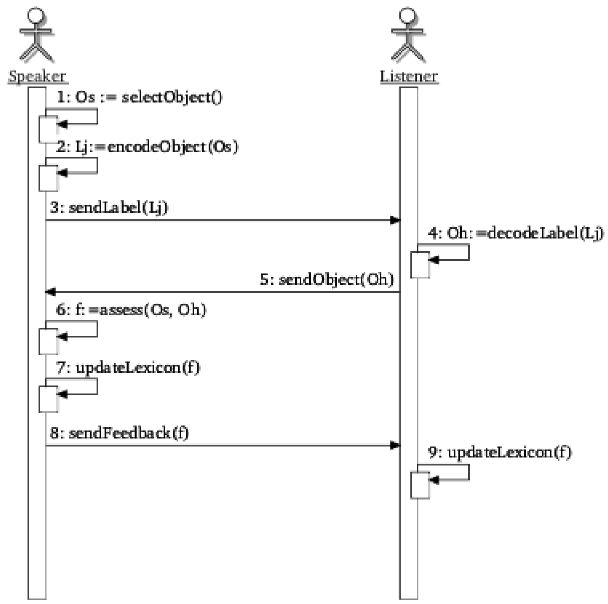

Figure 1: Language games interaction model.

relevance feedback to the agent that the solution of the retrieved case c_{i1} was not adequate: in our example we can see that the itinerary is not reviewed in sufficient detail. In response the agent looks up the case being inspected by the user in its association table. It finds that it corresponds to the lexical label l_1. It then issues a request for solutions for l_1 to the other agents that contain cases indexed as l_1. Each agent decodes the label l_1 and return cases associated with it. In the example shown in Figure 2 the solution for case c_{j9} from case base CB_j is returned to the user.

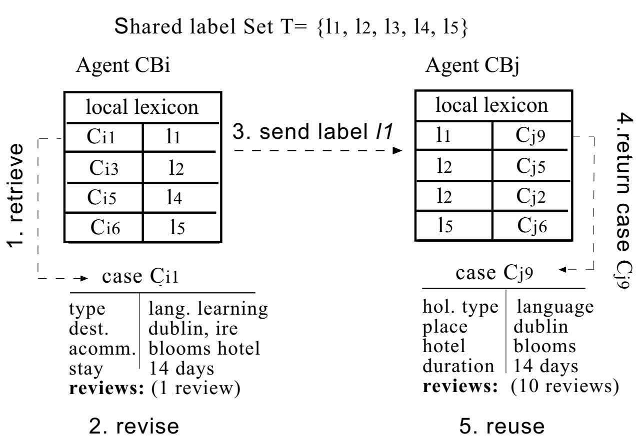

Figure 2: The retrieval model for distributed CBR agents.

5 Applications

5.1 Case-Based Information Agents

The previous section introduced the language game phenomenon as applied to a a community of distributed case-based agents. A detailed discussion of how language games solve the vocabulary alignment problem in distributed CBR can be found in [Avesani *et al.*, 2005b]. Figure 3 illustrates the results from an evaluation where we used a real data set from the Harmonise project, a European project to align services from heterogeneous tourism service providers [Fodor *et al.*, 2002]. To enable interoperability between different representations of the same event, the Harmonise researchers propose manually mapping each vendor's schema to an intermediate schema. Our evaluation goal was to see whether we could automatically align the same events using the language games methodology. Our data set consisted of 6 events represented four different ways by four case-based agents. Figure 3 illustrates that 100% alignment was achieved soon after 800 pairwise games.

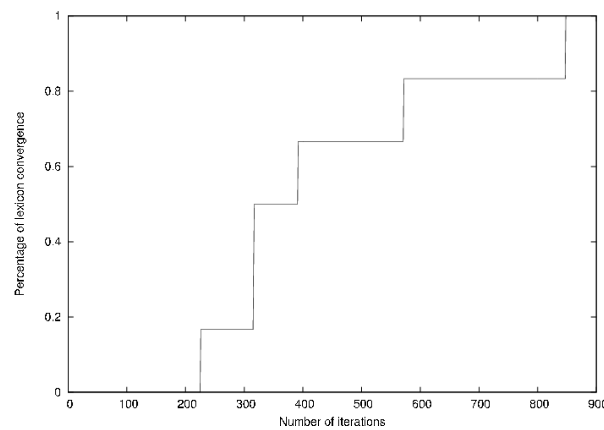

Figure 3: Formation of an indexing lexicon

5.2 Moleskiing

The Moleskiing initiative represents the authors' initial attemps at tackling the problem of the lack of implemented semantic standards on the web. In the sport of ski-mountaineering it is crucial for participants to have the most up to date reports on route conditions from other skiers. However, there are several ski-mountaineering portals to which alpine skiers can report conditions which tends to fragment the information available to other skiers. Moleskiing was designed as service to automatically reconcile the different schemas used by each portal to provide a single point of access on Alpine ski-mountaineering conditions. Three heterogeneous sources of ski-mountaineering information were used: Gulliver, Moleskiing and Skirando. Table 2 summarises the data from the three information providers that we used in our evaluation.

Figure 4 shows the plot of four sample game sessions. It shows the percentage of lexica convergence as a function of

	gulliver	*moleskiing*	*skirando*
Total items	38	179	69
gulliver overlap	-	22 (12%)	8 (11%)
moleskiing overlap	22 (57%)	-	51 (73%)
skirando overlap	8 (21%)	51 (28%)	-
Complete overlap	6 (15%)	6 (3%)	6 (8%)

Table 2: The lexicon of case base CB_j during the learning phase

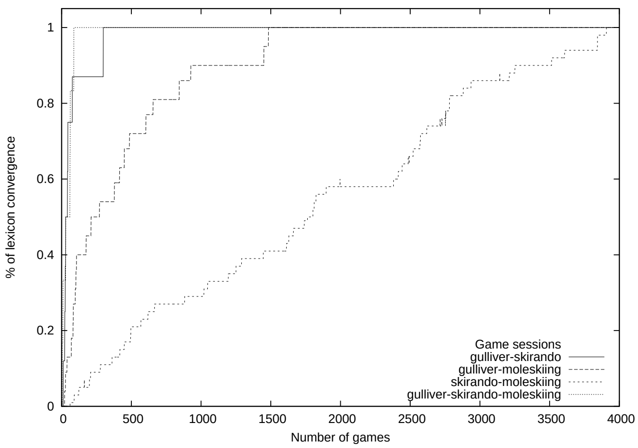

Figure 4: Evolution of common lexicon formation.

the number of games played between peer agents. 0% convergence means that there is no common associations among the peers: every peer is using different labels to encode the same object. Thus, the common lexicon is empty and inter-peer communication will fail. Conversely, 100% convergence indicates that peers have reached an agreement on how to reference all shared objects. The common lexicon contains one entry for each shared object and thus inter-peer communication is always successful. A full discussion of these results as well as a description of the service-oriented architecture for language games used in Moleskiing is given in [Avesani and Cova, 2005].

5.3 Tagsocratic

Weblogging has increasingly become an important part of the information economy found on the Internet [Nardi *et al.*, 2004; Schiano *et al.*, 2004]. Its great benefit is that it allows ordinary people to easily publish opinions based upon their experiences. This type of information, sometimes highly subjective, has great value for other Internet users who can make use of it to make decisions or simply to inform themselves. However, the blogging phenomenon exacerbates the problems posed by the lack of semantic protocols for the Internet. Although there is no constraint on what information can be posted, blogs often take the form of a series of annotations on topics of shared interest [Bar-Ilan, 2004]. As bloggers tend to publish their work independently, there is no standard way

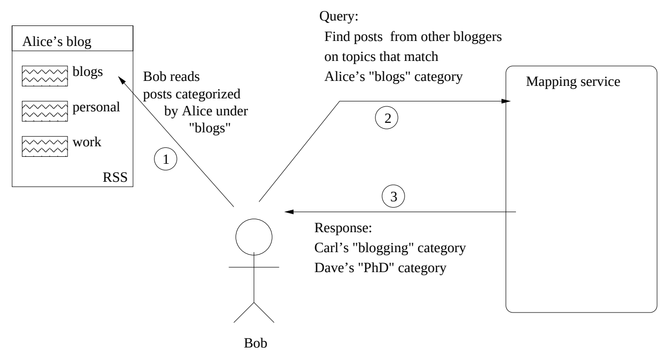

Figure 5: Tagsocratic use case.

of organising the blogosphere so that the posts that relate to a particular topic can be automatically indexed together. Most blog software allows users to define categories with which to label their posts. However, the semantics of the category are defined locally by the user rather than relating to a globally understood concept.

Clearly, there are benefits if these distributed information sources can be organised so that the reader (or blogger) can view related opinions on a single topic or concept. For example, the prominence given to user reviews on proprietary review sites like Amazon.com suggests their importance in providing sales advice to the potential customer. In Figure 5 we present a use case of the type of topic-centric service we require for the blogosphere. The objective is to provide an on-line mapping service for locally defined blog entry categories.

In the use case scenario depicted in Figure 5, Bob is visiting Alice's blog. He finds posts about the activity of blogging and notices that Alice categorises them under the category `blogs`. Bob would like to view other posts available in the blogosphere about the same topic. The problem, of course, is that other bloggers may use different categories to describe the *blogging* topic. Thus, Bob contacts the Tagsocratic service, requesting blog entries from categories mapped to Alice's `blogs` category; the mapping engine then returns a list of entries from categories aligned to the `blogs` category. The returned matches include entries from Carl under his `blogging` category and entries from Dave labelled `PhD` (Dave is doing a PhD on the effect of blogs on society). However, Bob does not receive any entries from Eve whose `blogs` category simply stores links to various blog engine web sites.

Thus, the objective of the Tagsocratic project is to provide an on-line matching service for local blog categories whilst respecting the autonomy of the blogger. Our goal is to allow a user to find posts categorised by other users under local categories that are semantically equivalent. In our approach, the semantics of other users' categories are automatically learnt by the system using the language games technique. The usage patterns of the user (which we call local context) are taken into account. This allows us, for example, to handle situations where two bloggers use the same category label with totally different meanings. From the functional point of view, Tagsocratic tackles the situation presented in the use case (see Figure 5). The issues involved in developing the Tagsocratic service are discussed in greater detail in [Avesani *et al.*, 2005a];

6 Future Work

We have begun examining how to reduce the number of game iterations required in order to converge on a stable lexicon. One area that can be improved is the initial period of game activity where the speaker sends a label which must be guessed by the listening agents. Clearly, many games must take place before the listener guesses correctly. One solution we have had initial success with is for the speaker to send an instance rather than a label. Secondly, we are examining the strategy used by the speaker in choosing labels or instances to send. Currently, these are chosen at random. However, it might be more efficient to send labels/instances that would provide more discriminating information to help in the formation of the local lexicon.

We have also begun to examine how different strategies can be employed by the peer agents. Currently, the strategy used by agents is naive: all agents are good potential game partners and partners are chosen at random for this purpose. However, we recognise that a more sophisticated strategy would be to choose partners that give the peer maximum information exposure (i.e. peer agents that service large communities) and partners that are consistently good sources of information. We describe some directions we are examining to allow agents to operate using more sophisticated strategies.

6.1 Mixed-initiative strategies

A mixed-initiative system is a system which allows more user interaction in the automated reasoning process. The key insight is that humans may be better equipped to assess critical points during the learning phase and should be enabled to contribute. By integrating the contributions from the user and system, we enable each to contribute what it does best. Moreover, flexible user-interaction policies would allow the system to adapt to differences in knowledge and preferences among users. In the context of language games research, we are investigating how a mixed-initiative strategy could be unobtrusively employed to speed up the convergence step by providing feedback on ambiguous lexical alignments during the learning phase. Furthermore, user interaction can help to narrow the scope of the game by selecting candidate players or barring further games with agents whose information services they distrust or dislike.

6.2 Trust/reputation strategies

The issue of trust and reputation on the Internet has become increasingly important, not just in terms of sales reliability but also in terms of implying consistency and authority [Richardson *et al.*, 2003]. For instance, Google's PageRank algorithm implicitly recognises highly linked pages as more likely to be authoritative sources of information [Page *et al.*, 1998]. Likewise, in the community of heterogenous information agents we have described, certain agents are likely to emerge as authorities on certain topic areas. Thus, rather than choosing partners at random, an agent may have more success in linking his topic descriptions to those expressed by such agents. We are interested in developing the protocols that allow reputation scores to be expressed and understood in a distributed environment.

6.3 Malicious recognition strategies

A related issue is how to recognise spam. At the present time, topic relevance is determined using classification and similarity matching techniques. However, spammers have shown ingenuity at gaming pattern recognition software and we would expect the determined spammer to be able to poison the distributed index with reference to non-relevant products and services. Our goal is to be able to detect spam early through a process of reputation metrics and pattern recognition.

6.4 Evaluation environment

A key issue is the evaluation of our system both in an offline and on-line context. There are three aspects to this: Firstly, we recognise that there is some correspondence between learning a distributed topic index and typical unsupervised learning techniques. The language games approach is novel in that it can be viewed as an unsupervised learning approach where the training corpus is decentralised. In terms of learning efficacy we are examining how we can formalise the language games approach so that it can be compared with typical centralised approaches to clustering.

Secondly, we are interested in developing a games simulator where we can test the languages technique using a game theoretic approach. Thus we can assign agents differing strategies and observe which strategies perform better. At the moment our objective function of game success is a global one indicating the percentage of global convergence. We recognise that we need to develop a more fine-grained function to measure overall game outcomes for individual agents and communities of agents.

Finally, we need to test our applications in an on-line setting. This is particularly important for evaluating the mixed-initiative strategy.

7 Conclusions

In this paper we investigated the problem of building the distributed common reference systems needed to enrich current web applications and allow for their meaningful interoperability. We considered this problem from the perspective of a community of distributed agents. Whereas the Semantic Web proposes a top-down approach to Semantic interoperability, we suggest that agents can learn the competences of other agents in their community. We described a novel approach to this problem based on the language games technique. We introduced three prototype applications we have developed to test this methodology: CBR agents, Moleskiing and the Tagsocratic project.

There is wide scope for future work. The model underlying the language games technique is still fairly unsophisticated and we plan to use the experience gained from practical experimentation to improve it. Along the same line, we expect to design more refined strategies to guide the games, in order to improve the lexicon building process.

Acknowledgments

This work was partially funded by the DiCA project, thanks to a grant from the (Istituto Nazionale Ricerca Montagna) and by the PeRec project funded by the PAT (Provincia Autonoma di Trento).

References

[Avesani and Agostini, 2003] P. Avesani and A. Agostini. A Peer-to-Peer Advertising Game. In M. Orlowksa, M. Papazoglou, S. Weerawarana, and J. Yang, editors, *First International Conference on Service Oriented Computing (ICSOC-03)*, pages 28–42, Berlin Heidelberg, 2003. Springer-Verlag LNCS 2910.

[Avesani and Cova, 2005] P. Avesani and M. Cova. Shared lexicon for distributed annotations on the web. In *Fourteenth International World Wide Web Conference*, Chiba, Japan, May 2005. Association for Computing Machinery (ACM).

[Avesani *et al.*, 2005a] P. Avesani, M. Cova, C. Hayes, and P. Massa. Learning contextualised weblog topics. Chiba, Japan, May 2005. WWW 2005 2nd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics.

[Avesani *et al.*, 2005b] P. Avesani, C. Hayes, and M. Cova. Language games: Solving the vocabulary problem in multi-case-base reasoning. The Sixth Internionial Conference on Case-Based Reasoning, Springer, August 2005.

[Bar-Ilan, 2004] J. Bar-Ilan. An outsider's view on 'topic-oriented blogging'. In *Proceedings of the 13th International World Wide Web Conference*, 2004.

[Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 279, May 2001.

[Finin *et al.*, 1994] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an Agent Communication Language. In N. Adam, B. Bhargava, and Y. Yesha, editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, Gaithersburg, MD, USA, 1994. ACM Press.

[FIPA, 2000] FIPA. Foundation for intelligent physical agents, 2000.

[Fodor *et al.*, 2002] O. Fodor, M. Dell'Erba, F. Ricci, and H. Werthner. Harmonise: a solution for data interoperability. In *Proceedings of the 2nd IFIP Conf.on E-Commerce, E-Business & E-Government*, Lisbon, Portugal, October 2002.

[Kuhn, 1955] H.W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistic Quarterly*, pages 83–97, 1955.

[Lewis, 1998] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, London, UK, 1998. Springer-Verlag.

[McGuinness *et al.*, 2002] D. L. McGuinness, R. Fikes, J.Hendler, and L. A. Stein. Daml+oil: An ontology language for the semantic web. *IEEE Intelligent Systems*, 17(5):72–80, 2002.

[Nardi *et al.*, 2004] B. A. Nardi, D. J. Schiano, and M. Gumbrecht. Blogging as social activity, or, would you let 900 million people read your diary? In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 222–231. ACM Press, 2004.

[Neches *et al.*, 1991] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. *AI Mag.*, 12(3):36–56, 1991.

[Page *et al.*, 1998] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[Rahm and Bernstein, 2001] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[Reed *et al.*, 2002] C. Reed, T. J. Norman, and N. R. Jennings. Negotiating the semantics of agent communication languages. *Computational Intelligence*, 18 (2):229–252, 2002.

[Richardson *et al.*, 2003] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *International Semantic Web Conference*, pages 351–368, 2003.

[Schiano *et al.*, 2004] D. J. Schiano, B. A. Nardi, M. Gumbrecht, and L. Swartz. Blogging by the rest of us. In *Extended abstracts of the 2004 conference on Human factors and computing systems*, pages 1143–1146. ACM Press, 2004.

[Steels and McIntyre, 1999] L. Steels and A. McIntyre. Spatially Distributed Naming Games. *Advances in Complex Systems*, 1(4):301–323, January 1999.