Measuring Data-Driven Ontology Changes using Text Mining

Majigsuren Enkhsaikhan **Wilson Wong** **Wei Liu†** **Mark Reynolds**

School of Computer Science and Software Engineering
University of Western Australia
35 Stirling Highway, Crawley, WA 6009
E-mail: majigaa,wilson,wei,mark}@csse.uwa.edu.au

† Corresponding author.

Abstract

Most current ontology management systems concentrate on detecting usage-driven changes and representing changes formally in order to maintain the consistency. In this paper, we present a semi-automatic approach for measuring and visualising data-driven changes through ontology learning. Terms are first generated using text mining techniques using an ontology learning module, and then classified automatically into clusters. The clusters are then manually named, which is the only manual process in this system. Each cluster is considered as a sub-concept of the root concept, and thus one dimension of the feature space describing the root concept. The changes of terms in each cluster contributes to the change of the root concept. Using our system, Web documents are collected at different time periods and fed into the system to generate different versions of the same ontology for each time period. The paper presents several ways of visualising and analysing the changes. Initial experiments on online media data have demonstrated the promising capabilities of our system.

1 Introduction

An ontology for a dynamic domain is constantly evolving. It changes over time as the underlying knowledge fluctuates. Ontology changes also occur when human and software agents modify an ontology when applying them in either a centralised or distributed environment. (Flouris 2006) identified nine different tasks related to ontology change to summarise the state of the art. These include ontology mapping (Ehrig & Sure 2004), morphism (Kalfoglou & Schorlemmer 2005), alignment (Euzenat et al. 2004, de Bruijn et al. 2004), articulation, translation, evolution (Stojanovic 2004), versioning (Stojanovic 2004), integration (Pinto & Martins 2001) and merging (de Bruijn et al. 2004, Hitzler et al. 2005). Ontology mapping, morphism, alignment, articulation are to resolve the heterogeneities between ontologies to enable interoperability, while translation, evolution, versioning are to maintain the consistency and integrity of a single ontology in the face of changes, merging and integration are to unify all concepts and relations of source ontologies to suit the domain needs.

In this paper, we are interested in ontology evolution, which is defined as the timely adaptation to the changes in the business requirements, in the usage of the ontology-based applications, and in the modifications of consistent management of the changes (Stojanovic 2004). Changes may generate other changes because of the relations of vocabularies or axioms in a single ontology or the relations of several ontologies. Inconsistencies in the same or dependent ontologies can be also generated. Therefore, ontology evolution can become a very complex operation to maintain the consistency without losing information. In this paper, we simplify the notion of ontology evolution to only look at the change of vocabularies or terminologies over time in certain domains such that we can identify the changes and determine the amount of changes automatically.

Detecting and measuring ontology change over time is often done by manual inspection of different versions of the ontology before and after the change, or keeping change logs to record the modifications. In the case of ontologies generated automatically or semi-automatically through text mining (Liu et al. 2005), it is possible to automatically detect the changes and the degree of changes by comparing the ontologies generated for different time periods.

This paper describes a text mining approach for ranking domain terms and generating term clusters to measure the amount of changes in certain domains. Section 2 provides the background and relevant work in ontology evolution. Section 3 presents the techniques used in building different modules of our system. Section 4 discusses the experiments and results. The paper concludes with an outlook to future work in Section 5.

2 Related Work

2.1 Process of ontology evolution

According to (Stojanovic 2004), the process of ontology evolution has 6 cyclic phases:*Change Capturing*, *Change Representation*, *Semantics of change*, *Change implementation*, *Change propagation*, *Change validation*.

The approach proposed in this paper deals with the change capturing phase only. Changes can be captured from explicit requirements or implicit requirements. Explicit requirements generated by ontology engineers or end users are defined as *top-down changes*. Changes from implicit requirements like change discovery methods are called *bottom-up changes*. Implicit changes are categorized in three types: *structure-driven* change from ontology structure, *usage-driven* change from usage patterns created over time and *data-driven* change from modification of underlying knowledge such as text files. Specifically, while Stojanovic et. al. (Stojanovic 2004) discuss how to keep and mine the usage logs to discover usage driven changes, here we are interested in the data-driven changes that are hidden in large domain corpora, which is constantly modified and updated by adding new documents or deleting obsolete documents.

2.2 Formalisms for change representation

A change log records an exact sequence of changes that occurred when an ontology developer updated an old version of an ontology to a new version. If the recording of a change log is unavailable in a dynamic and distributed environment like the Semantic Web, then old and new versions of an ontology bring the possibility to define changes (Klein 2004, Plessers et al. 2007). To declare changes from old and new versions of the same ontology, we can use the following techniques: *structural difference*, *conceptual change* and *transformation set*. Structural difference is a map of correspondences between old and new versions of the ontology. It gives the declarative view of ontology transition by comparing the versions of the ontology. Conceptual changes between old and new versions identify changes in the concepts. Transformation set provides a set of change operations that specify how an old ontology version can be transformed into a new version. It lists the changes as a result of comparison of two versions of the ontology.

Preserving history of an ontology is used for creating ontology versions. There are two ways to preserve history of an ontology. *Timestamp* approach labels elements of an ontology with specific time. *Snapshot* approach applies snapshots to capture the different states of an ontology over time. Snapshots can be for the whole ontology or for a single concept definition (Plessers 2006).

In this paper, we opt to the snapshot approach. At each time period, an ontology consisting of concepts and relations is kept. The same ontology at different time period populates an ontology repository.

2.3 Existing Ontology Evolution Systems

Existing ontology evolution systems are centered around two dominant ontology management infrastructures, the KArlsruhe ONtology and Semantic Web Framework (KAON) [1] and Protégé [2]. They both provide a suite of tools or plug-ins for creating, editing, modifying and reasoning about ontologies for Semantic Web applications.

Stojanovic (Stojanovic 2004) detects usage-driven changes by analysing usage patterns and query logs. Haase et. al. (Haase et al. 2004, Haase & Sure 2005)'s system to maintain consistency and discover changes during the usage of an ontology based application (in this case, an online digital library). Flouris et. al. (Flouris et al. 2006) proposed automatic ontology evolution based on belief change theory. Plessers et. al. (Plessers et al. 2007) developed *Change Definition Language* to facilitate formal representation and reasoning about changes. Noy et. al. (Noy et al. 2006) developed an ontology evolution framework in a collaborative environment. It supports different modes of ontology changes in a single framework by using several formalisms of change representation.

Overall, detecting usage-driven changes and maintaining consistency through reasoning about changes and formal representation are the main concerns of the current systems. Our system complements the existing ones on dealing with data-driven changes.

3 Change Detection using an Ontology Learning System

Figure 1 shows an overview of our system, consisting of an ontology learning module, a visualisation module and various sub-modules. Text collected from the Web is first pre-processed to remove spelling errors and resolve acronyms. Clean text is then fed into term extraction and concept discovery sub-modules to generate domain relevant terms and group relevant ones into clusters. The clusters are then named manually by domain experts. The clusters and the weights of the domain terms are then used in the visualisation module to produce graphs and tables to aid human comprehension of the results.

[1] http://kaon.semanticweb.org/
[2] http://protege.stanford.edu/

The main building blocks of domain ontologies are domain-specific concepts. Since concepts are merely mental symbols that we employ to represent the different aspects of a domain, they can never really be computationally captured from written or spoken resources. Instead, in ontology learning, terms are regarded as lexical realisations for expressing or representing concepts that characterise various aspects of specialised domains. Therefore, hereafter, we use terms and concepts interchangeably.

3.1 Term Extraction and Concept Discovery

The main task in term extraction is to determine whether a word or phrase is a term that characterises the target domain. This key question can be further decomposed to reveal two critical notions in this area, namely, *unithood* and *termhood*. Unithood concerns whether sequences of words should be combined to form stable lexical units, while termhood is the degree to which these stable lexical units are relevant to some domains. After processing the domain text using a linguistic parser and unithood analysis, Wong et al. (2007*b*,*a*) presented an empirically-derived scoring and ranking scheme for the determination of termhood based on a set of heuristically-motivated term characteristics. The scoring scheme ranks term candidates using numerical weights, indicating the significance of the concepts in the domain. Therefore, here we give a brief explanation of how we obtained the weights.

Two base measures are introduced for capturing the statistical evidence based on the cross-domain and intra-domain distribution of term candidates and their context words, respectively:

- *Domain Prevalence (DP)*, to measure the extent of term usage in the domain.
- *Domain Tendency (DT)*, to measure the extent of inclination of term usage in the domain.

A high *DP* means that the term is frequently encountered (i.e. prevalent) in the target domain. It is a sign of high domain relevance if and only if the frequent usage of that term is exclusive to the target domain, that is, has high domain tendency DT. These two measures are further adjusted when taking into account three types of linguistic evidence. Namely, **Candidate evidence** as *discriminative weight (DW)*, **Modifier evidence** as *modifier factor (MF)* and **Contextual evidence** as *average contextual discriminative weight (ACDW)*.

This new mechanism requires a corpus containing text generated using the special language of the target domain (i.e. domain corpus) and a set of corpora produced using special languages from domains other than the target domain (i.e. contrastive corpora). Details of how to obtain the above measures are presented below:

Given that we have a list of term candidates (both simple and complex) $TC = \{a_i, ..., a_n\}$, to determine termhood is to assign weights to term candidates in order to identify the m most suitable candidates as terms in a domain $t \in T$. Each complex term, a will comprise of a head a^h and modifiers $m \in M(a)$. Each term candidate is assigned a weight depending on its type (i.e. simple or complex). Inspired by the contrastive weights (CW)by Basili et. al. (Basili, Moschitti, Pazienza & Zanzotto 2001), the *domain prevalence (DP)* for a simple term a is defined as:

$$DP(a) = \log_{10}(f_{ad} + 10)\log_{10}\left(\frac{F_{TC}}{f_{ad} + f_{a\bar{d}}} + 10\right)$$

where $F_{TC} = \sum_j f_{jd} + \sum_j f_{j\bar{d}}$ is the sum of the frequencies of occurrences of all $a \in TC$ in both domain and contrastive corpora, while f_{ad} and $f_{a\bar{d}}$ are the frequencies of occurrences of a in the domain corpus and contrastive corpora, respectively. If the term candidate is complex, we

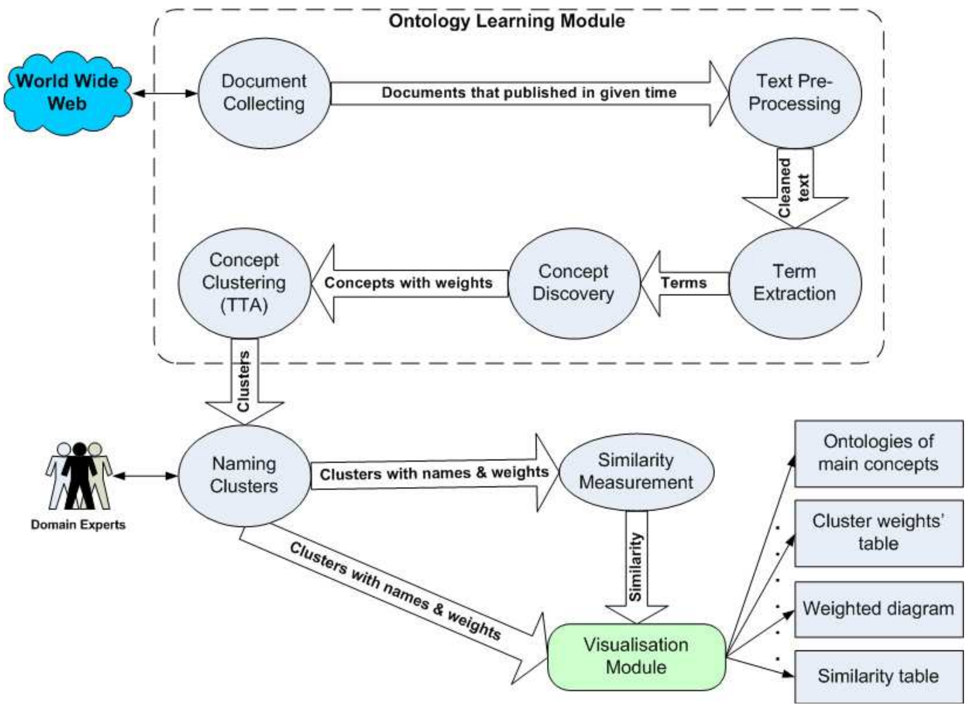

Figure 1: Architecture for Detecting Changes using an Ontology Learning System

define its *DP* as:

$$DP(a) = \log_{10}(f_{ad} + 10)DP(a^h)MF(a)$$

Based on our preliminary experiments on comparing *DP* with the original *CW*, to address the biased ranking by *CW*, we add a constant 10 to f_{ad} prior to log and introduce another new measure called *modifier factor (MF)*. The *MF* of a complex term *a* is measured using the cumulative domain frequency and cumulative contrastive frequency of modifiers which also happen to be term candidates, $m \in M_a \cap TC$. Formally, the *MF* of a complex term *a* is defined as:

$$MF(a) = \log_2 \left(\frac{\sum_{m \in M_a \cap TC} f_{md} + 1}{\sum_{m \in M_a \cap TC} f_{m\bar{d}} + 1} + 1 \right)$$

MF is actually a derived measure modelled after our second new base measure *domain tendency (DT)*. The only difference is that *MF* works with modifiers while *DT* works with the entire term candidate, both simple and complex. Formally, we can determine the inclination of using term candidate *a* for domain and non-domain purposes through:

$$DT(a) = \log_2 \left(\frac{f_{ad} + 1}{f_{a\bar{d}} + 1} + 1 \right)$$

If term candidate *a* is equally common in both domain and non-domain (i.e. contrastive domain), $DT = 1$. If the usage of *a* is more inclined toward the target domain, $f_{ad} > f_{a\bar{d}}$, then $DT > 1$, and $DT < 1$ otherwise. Next, this new base measure *DT* together with *DP* will contribute to a new weight known as *discriminative weight (DW)*, which is simply the product of *DP* and the corresponding *DT* of the term candidate:

$$DW(a) = DP(a)DT(a)$$

Assuming that term candidate *a* has the set of context words C_a, the *average contextual discriminative weight (ACDW)* is defined as:

$$ACDW(a) = \frac{\sum_{c \in C_a} DW(c)NGD(a,c)}{|C_a|}$$

where $NGD(a,c)$ is the normalized google distance (Cilibrasi & Vitanyi 2006) between term candidate *a* and *c*. The *ACDW* weight allows us to take into consideration the company a term candidate keeps. Nonetheless, not all context words describe or are related to the terms they appear with. Therefore, we adjust *ACDW* according to its ratio with the corresponding *DW* to produce the *adjusted contextual contribution (ACC)* as:

$$ACC(a) = ACDW(a) \frac{e^{\left(1 - \frac{ACDW(a)+1}{DW(a)+1}\right)} e^{\left(1 - \frac{DW(a)+1}{ACDW(a)+1}\right)}}{\log_2 \frac{ACDW(a)+1}{DW(a)+1} + 1}$$

In the end, we define the final weight known as *termhood (TH)* for each term candidate as:

$$TH(a) = DW(a) + ACC(a) \tag{1}$$

3.2 Concept clustering

Once the term candidates have been scored and ranked, filtering and selection are usually performed using some thresholds or with minimal expert involvements. The result is a list of terms which has been deemed fit to denote some domain-specific concepts or is relevant to certain domains of interest. As shown in Figure 1, these terms are the ones that will contribute to concept discovery. Wong et al. (2006, 2007*c*) presented a novel clustering algorithm known as the *Tree-Traversing Ant (TTA)* for discovering concepts from terms. *TTA* was designed as an attempt to fuse the strengths of standard ant-based methods with certain advantages of conventional clustering methods. One of the biggest issues associated with term clustering is the lack of visible (e.g. physical and behavioral) features required for the computation of similarity. Together with the use of featureless similarity measures, namely *Normalised Google Distance (NGD)* (Cilibrasi & Vitanyi 2006) and *n° of Wikipedia (n°W)* (Wong et al. 2007*c*), *TTA* was able to address the issues related to similarity measurement and many other unique requirements of term clustering in ontology learning. Seven of the most notable strengths of *TTA* with respect to clustering are:

- Able to further distinguish hidden structures within clusters;
- Flexible in regards to the discovery of clusters;
- Capable of identifying and isolating outliers;
- Tolerance to differing cluster sizes;
- Able to produce consistent results;
- Able to identify implicit taxonomic relationships between clusters; and
- Inherent capability of coping with synonyms, word senses and the fluctuation in terms usage.

3.3 Measuring the similarities between concepts

Techniques for measuring the similarities between concepts can employ extensive and well-ground semantic resources such as WordNet, OpenCyc or domain specific ontologies to compute the distance between two concepts. For example, (Maynard & Ananiadou 1999, 2000) consult the *Unified Medical Language System (UMLS)* to compute two weights, namely, positional and commonality in order to measure the similarity between two concepts. Positional weight is obtained based on the combined number of nodes belonging to each word, while commonality is measured by the number of shared common ancestors multiplied by the number of words. Accordingly, the similarity between two term candidates is defined as:

$$sim(a,b) = \frac{com(a,b)}{pos(a,b)} \tag{2}$$

where $com(a,b)$ and $pos(a,b)$ is the commonality and positional weight respectively, between term candidate a and b.

However, such an approach is not viable when there is no ready to use domain ontology or taxonomy available. In addition, Basili et. al. (Basili, Pazienza & Zanzotto 2001) criticised that such approach for being so reliant on existing ontologies and thus not portable to other domains. Therefore, they combine the use of contextual information and the head-modifier principle to capture term candidates and their context words on a feature space for computing similarity. Given the term candidate a, the feature vector for a is:

$$\tau(a) = (f_1, ..., f_n)$$

where f_i is the value of the attribute F_i and n is the number of features. The authors chose cosine measure for computing similarity over the syntactic feature space:

$$sim(\tau_i, \tau_j) = \frac{\tau_i \tau_j}{|\tau_i||\tau_j|}$$

This approach require large corpora and high density of the domain terms to be effective. To ensure the general applicability of our approach, the cosine similarity measure is more suitable for us in measuring the similarities between ontologies generated at different periods of time. This will free us from assuming prior ontologies. It is also feasible because our system is capable of collecting large volumes of online text data to ensure an optimal size of the corpora. As you will see in Section 4, a general root concept is represented using a vector of sub-domains (or clusters), the children concepts in each sub-domain contributes to the overall weight of the sub-domain. Therefore, changes in the significance of each sub-domain measured by the corresponding overall weight can be used in measuring the shift in the semantics of the root concept. Cosine similarity measure is used to compute the amount of changes over a specified time period.

4 Experiments and Results

The *document collecting* sub-module as shown in Figure 1 reads the RSS feed from major Australian online News outlets every day from September 2006 to April 2007. We choose to use online news articles to demonstrate the capability of our system because media data are more volatile than other types of domain corpora. Changes are therefore more obvious.

As online media articles in Australia cover news in Australia as well as international news that made their way into Australia, we treat the terms and relations extracted using the ontology learning module as features of a general root concept `Australia`. As far as the online media domain is concerned, `Australia` is a constantly evolving concept. In our case, the changes are represented by the changes in its features, that is, terms and relations.

As shown in Figure 2-4, three versions of the ontology for the root concept `Australia` in the media domain are generated for September 2006, December 2006 and April 2007. The generation of terms and the creation of links are done automatically using the termhood measurement TH and Tree Traversing Ants algorithm discussed in Section 3. Unfortunately, as it is still an open research question we are currently pursuing, there is not yet means of automatically naming the clusters in our current system. Manual input from domain experts is therefore necessary in determining the name of the clusters based on the common node shared by the terms. Identifying the cluster is important in our approach of measuring the amount of changes, because each cluster is considered as a sub-domain and therefore a dimension in the feature space of the general root concept. In our case, the concept `Australia` in the media domain is considered represented by a vector with 7 features, namely, `Politics`, `Economy`, `Policies`, `Environmental Issues`, `Domestic Affairs` and `International Affairs`. However, as one will notice that some anomaly might exist. For example, `poor housing affordability` ideally should be classified in the domestic affairs cluster. This issue was discussed in (Wong et al. 2006, 2007*c*), the infrequent occurrences of such anomalies in a semi-automatic system is tolerable.

The *visualisation module* in Figure 1 allows different types of plug-ins to display the changes in sensible ways to ease the comprehension and interpretation of the changes. To quantify the changes and to weigh the importance of each cluster in the general root concept, we defined the accumulative weight (ω) of each cluster as the sum of the TH value of each term in the cluster,

$$\omega_i = \textstyle\sum_{k=1}^{n} (TH_k)$$

where n is the number of terms in cluster i.

Figure 5 represents these accumulative weights of each cluster for all three periods. The data can be better visualised using scaled circles to indicate the amount of media attention in certain feature dimensions, as shown in Figure 6.

It is not surprising to see that media attention to the main topic areas in Australia are constantly changing. Figure 6 is very intuitive in showing the volatile clusters and the relative stable clusters. From here, it is obvious that some areas, such as coverage on `Environmental issues`, `Sports and Entertainment` and `Economy`, are changing quite dramatically over different periods, while `Politics` and `Domestic Affairs`, are relatively stable. The steady increase in the amount politics coverage from September 2006, to December 2006 and April 2007 indeed makes sense, as one would expect more talking-about on politics as the election draws close.

Overall changes in such an ontology are generated by the changes of internal terms. Looking into the terms comprising of each cluster, the topic changes depend on many aspects such as sudden events, sports season, weather and

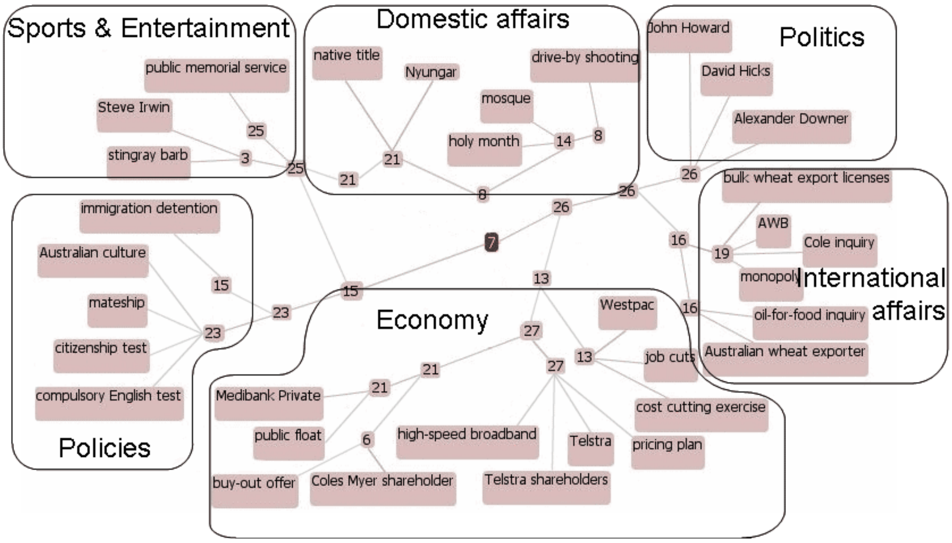

Figure 2: Clusters of September 2006

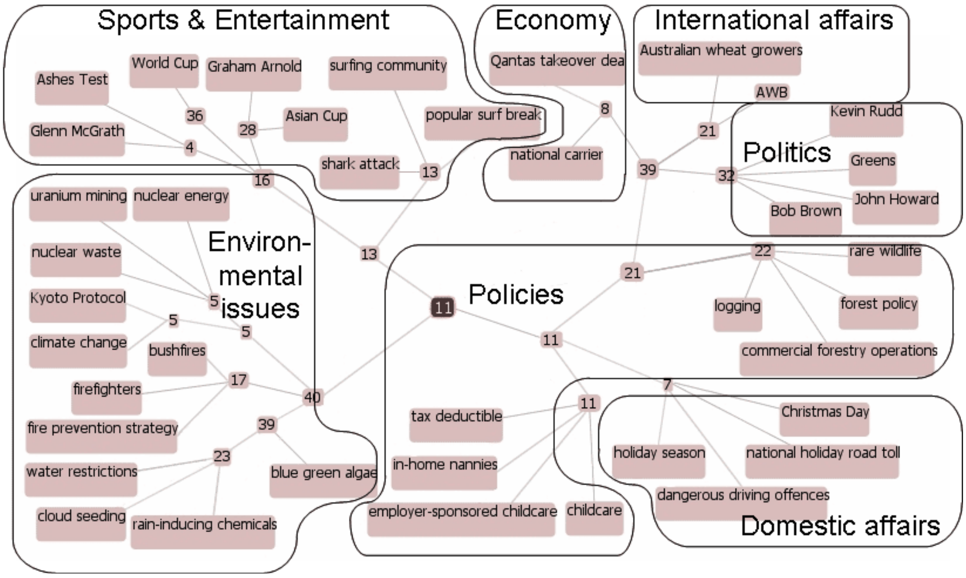

Figure 3: Clusters of December 2006

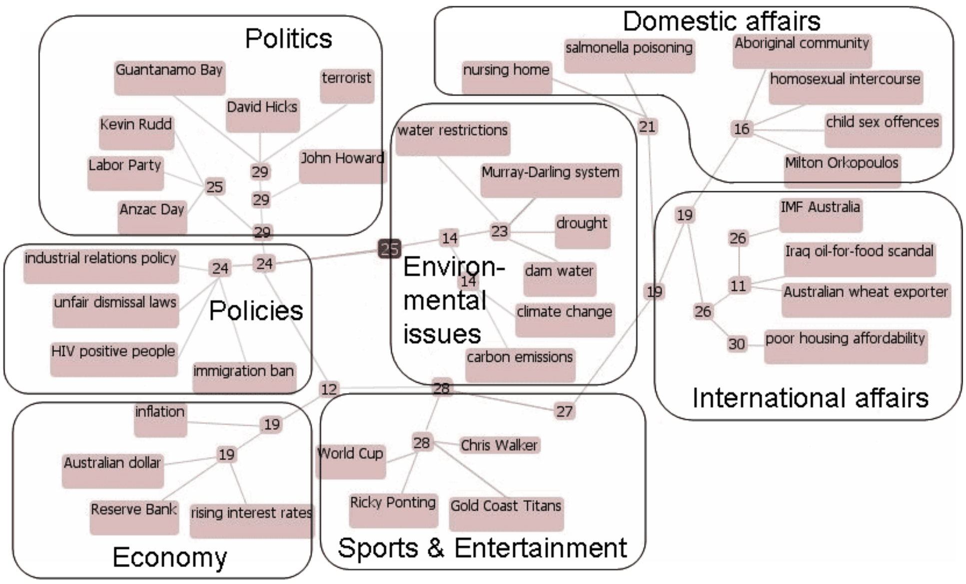

Figure 4: Clusters of April 2007

Period \ Cluster	Politics	Sports & Entertainment	Economy	Environmental issues	International affairs	Domestic affairs	Policies
April 2007	120.37	60.81	52.86	80.3	49.68	91.54	59.83
December 2006	90.75	128.93	28.9	105.17	33.09	47.3	107
September 2006	69.96	54.54	161.08	0	92.09	72.17	4

Figure 5: Cluster weights

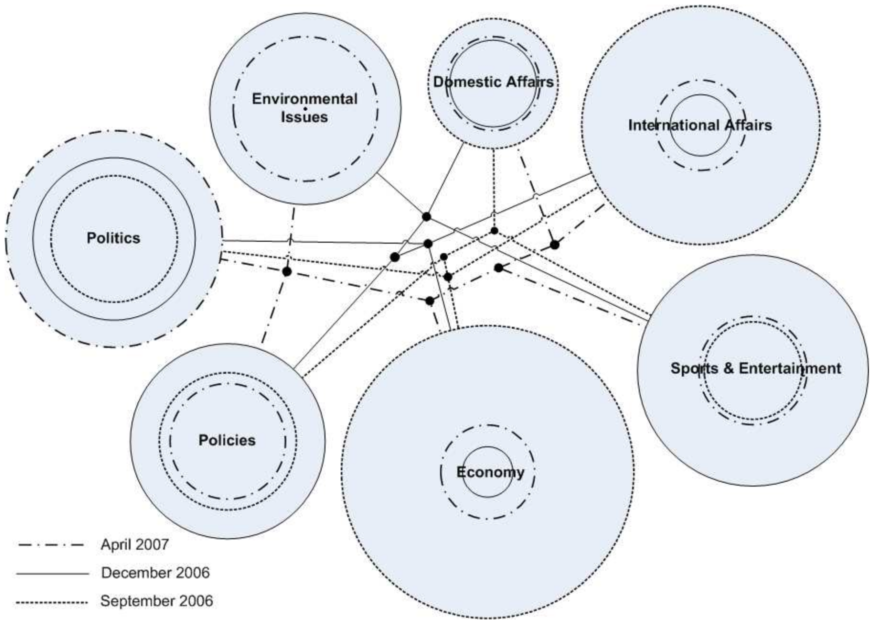

Figure 6: Clusters in all three periods

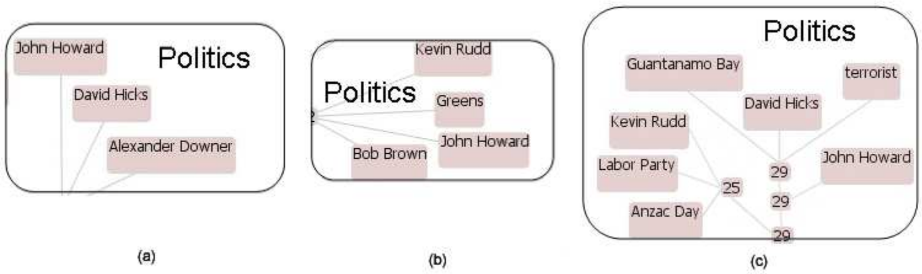

Figure 7: Politics area: (a) in September 2006; (b) in December 2006; (c) in April 2007

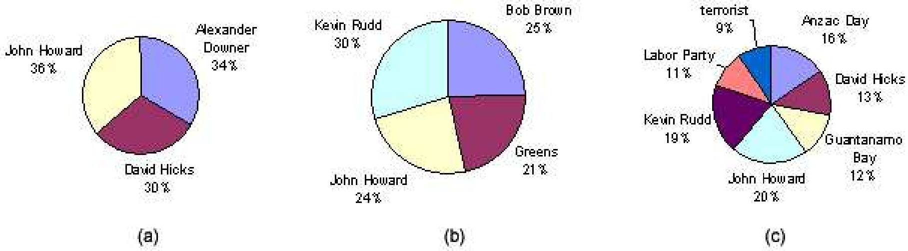

Figure 8: Term contributions in Politics: (a) in September 2006; (b) in December 2006; (c) in April 2007

Time periods	April 2007	December 2006	September 2006
April 2007	x	0.627	0.762
December 2006	0.627	x	0.622
September 2006	0.762	0.622	x

Figure 9: Similarity measurement

etc. In December 2006, which was a hot summer in Australia, `bushfires` appeared and environmental issues became one of the hot topics, even though they were not strongly mentioned during September 2006. Taken the `Politics` cluster for example, Figure 7 zoom into this cluster in the three different periods. As we can see, all three of them includes the Prime Minister of Australia, Mr. John Howard, as one of the main terms. Other terms differed and did not appear in all of them. Before becoming the new Opposition leader, Mr. Kevin Rudd was not in the cluster in September 2006, but appeared in December 2006 and came stronger in significance (TH value) after he took the position. Therefore, measuring the contribution from each terms to the cluster can also be important. As shown in Figure 8(b), `Kevin Rudd` contributes 30% to the cluster as compared to `John Howard`'s 24%. The overall changes in the ontology over different time periods can be measured using the cosine similarity of the vectors representing each period. Given two versions of the ontology at two different time period, O_t and $O_t^{'}$, the vectors are $\tau = (\omega_1, ..., \omega_m)$ and $\tau^{'} = (\omega_1^{'}, ..., \omega_m^{'})$, where m is the number of features. According to the formula in Subsection 3.3, we have

$$sim(O_t, O_t^{'}) = \frac{\sum_{i=1}^{n}(\omega_i \omega_i^{'})}{\sqrt{\sum_{i=1}^{n}(\omega_i)^2}\sqrt{\sum_{i=1}^{n}(\omega_i^{'})^2}}$$

where n is the number of clusters in the ontology, ω_i is the cumulative weight for a selected cluster. A $sim(O_t, O_t^{'})$ value closer to 1 indicates higher similarity, whereas a value closer to 0 signifies higher distance. We measured the cosine similarity of every two different versions for the general root concept `Australia`. The value varied between 0.62 and 0.76 in our selected time periods as shown in Figure 9.

From here, we can say that the concept `Australia` fluctuates 24%$\sim$38% between September 2006 and April 2007.

5 Conclusion and Future Work

Data driven ontology changes can be detected using text mining based ontology learning systems. In this paper, we presented an approach for semi-automatically generating ontology concept clusters at different time periods, measuring and visualising the changes in sensible ways to help understand the overall concept changes as well as the individual terms contributed to the change. Experiments have confirmed the effectiveness and intuitiveness of the system. The results are also quite sensible in explaining real world events.

As the data source is from volatile online media domain, the generated ontology shows little consistency at the term level. However, this can be helped by grouping the relevant terms into clusters, which are relatively stable over the time span considered. In the future, comparison experiments are planned on more domain specific text to see if the same approach can be used to discover general or upper ontology.

References

Basili, R., Moschitti, A., Pazienza, M. & Zanzotto, F. (2001), A contrastive approach to term extraction, *in* 'Proceedings of the 4th Terminology and Artificial Intelligence Conference (TIA)', France.

Basili, R., Pazienza, M. & Zanzotto, F. (2001), Modelling syntactic context in automatic term extraction, *in* 'Proceedings of the International Conference on Recent Advances in Natural Language Processing', Bulgaria.

Cilibrasi, R. & Vitanyi, P. (2006), Automatic extraction of meaning from the web, *in* 'Proceedings of the IEEE International Symposium on Information Theory', Seattle, USA.

de Bruijn, J., Martin-Recuerda, F., Manov, D. & Ehrig, M. (2004), State-of-the-art survey on ontology merging and aligning, Sekt deliverable d4.2.1, Digital Enterprise Research Institute, University of Innsbruck.

Ehrig, M. & Sure, Y. (2004), Ontology mapping - an integrated approach, *in* C. Bussler, J. Davis, D. Fensel & R. Studer, eds, 'Proceedings of the 1st European Semantic Web Symposium (ESWS 2004) on The Semantic Web: Research and Applications', Vol. 3053 of *Lecture Notes in Computer Science*, Springer, pp. 76–91.

Euzenat, J., Bach, T. & J. Barrasa, P. Bouquet, e. a. (2004), State of the art on ontology alignment, Knowledge web deliverable, Knowledge Web Consortium.

Flouris, G. (2006), On Belief Change and Ontology Evolution, PhD thesis, University of Crete, Greece.

Flouris, G., Plexousakis, D. & Antoniou, G. (2006), Evolving ontology evolution, *in* 'Proceedings of the 32nd International Conference on Current Trend in Theory and Practice of Computer Science (SOFSEM-2006)'.

Haase, P. & Sure, Y. (2005), Incremental ontology evolution - evaluation, Sekt deliverable d3.1.2, Institute AIFB, University of Karlsruhe.

Haase, P., Sure, Y. & Vrandecic, D. (2004), Ontology management and evolution - survey, methods and prototypes, Sekt deliverable d3.1.1, Institute AIFB, University of Karlsruhe.

Hitzler, P., Krotzsch, M., Ehrig, M. & Sure, Y. (2005), What is ontology merging? - a category-theoretical perspective using pushouts.

Kalfoglou, Y. & Schorlemmer, M. (2005), Ontology mapping: The state of the art, *in* Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab & M. Uschold, eds, 'Semantic Interoperability and Integration', number 04391 *in* 'Dagstuhl Seminar Proceedings'.

Klein, M. (2004), Change Management for Distributed Ontologies, PhD thesis, Vrije Universiteit Amsterdam.

Liu, W., Weichselbraun, A., Scharl, A. & Chang, E. (2005), 'Semi-automatic ontology extension using spreading activation', *Journal of Universal Knowledge Management* **0**(1), 50–58.

Maynard, D. & Ananiadou, S. (1999), Term extraction using a similarity-based approach, *in* 'Recent Advances in Computational Terminology', John Benjamins.

Maynard, D. & Ananiadou, S. (2000), Identifying terms by their family and friends, *in* 'Proceedings of the 18th International Conference on Computational Linguistics', Germany.

Noy, N., Chugh, A., Liu, W. & Musen, M. (2006), A framework for ontology evolution in collaborative environments, *in* 'Proceedings of the 5th International Semantic Web Conference (ISWC-2006)', Vol. 4273 of *Lecture Notes in Computer Science*.

Pinto, H. & Martins, J. (2001), Ontology integration: How to perform the process, *in* 'Proceedings of IJCAI2001 Workshop on Ontologies and Information Sharing', AAAI Press, pp. 71–80.

Plessers, P. (2006), An Approach to Web-Based Ontology Evolution, PhD thesis, Vrije Universiteit Brussel.

Plessers, P., Troyer, O. D. & Casteleyn, S. (2007), 'Understanding ontology evolution: A change detection approach', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **5**, 39–49.

Stojanovic, L. (2004), Methods and Tools for Ontology Evolution, PhD thesis, University of Karlsruhe.

Wong, W., Liu, W. & Bennamoun, M. (2006), Terms clustering using tree-traversing ants and featureless similarities, *in* 'Proceedings of the International Symposium on Practical Cognitive Agents and Robots', Perth, Australia.

Wong, W., Liu, W. & Bennamoun, M. (2007*a*), Determining termhood for learning domain ontologies in a probabilistic framework, *in* 'Proceedings of the 6th Australasian Conference on Data Mining (AusDM)', Gold Coast.

Wong, W., Liu, W. & Bennamoun, M. (2007*b*), Determining the unithood of word sequences using mutual information and independence measure, *in* 'Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING)', Melbourne, Australia.

Wong, W., Liu, W. & Bennamoun, M. (2007*c*), 'Tree-traversing ant algorithm for term clustering based on featureless similarities', Journal on Data Mining and Knowledge Discovery, doi: 10.1007/s10618-007-0073-y.