The Knowledge Engineering Review, Vol. 31:2, 97–123.
doi:10.1017/S0269888916000047

Data mining for building knowledge bases: techniques, architectures and applications

ALFRED KRZYWICKI, WAYNE WOBCKE, MICHAEL BAIN,
JOHN CALVO MARTINEZ and PAUL COMPTON

School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia;
e-mail: alfredk@cse.unsw.edu.au, wobcke@cse.unsw.edu.au, mike@cse.unsw.edu.au, jcalvo@cse.unsw.edu.au,
compton@cse.unsw.edu.au

Abstract

Data mining techniques for extracting knowledge from text have been applied extensively to applications including question answering, document summarisation, event extraction and trend monitoring. However, current methods have mainly been tested on small-scale customised data sets for specific purposes. The availability of large volumes of data and high-velocity data streams (such as social media feeds) motivates the need to automatically extract knowledge from such data sources and to generalise existing approaches to more practical applications. Recently, several architectures have been proposed for what we call *knowledge mining*: integrating data mining for knowledge extraction from unstructured text (possibly making use of a knowledge base), and at the same time, consistently incorporating this new information into the knowledge base. After describing a number of existing knowledge mining systems, we review the state-of-the-art literature on both current text mining methods (emphasising stream mining) and techniques for the construction and maintenance of knowledge bases. In particular, we focus on mining entities and relations from unstructured text data sources, entity disambiguation, entity linking and question answering. We conclude by highlighting general trends in knowledge mining research and identifying problems that require further research to enable more extensive use of knowledge bases.

1 Introduction

Human knowledge is distributed among a very large number of sources, each with constantly growing volumes of data. Knowledge bases (KBs) are becoming indispensible tools for automatic understanding of human-generated data, summarisation and question answering. In the past, a KB was a body of knowledge included in an Expert System, commonly expressed as a collection of rules, limited to a specific domain and for a specific purpose, for example blood analysis for medical diagnosis. In the last 15 years, however, there has been a growing trend to construct KBs that encompass broader domains, crossing domain boundaries and becoming large summaries of general knowledge.

In this survey, we provide a review of the state-of-the-art literature on integrated data mining techniques for the construction, maintenance and use of KBs, with an emphasis on using fast unstructured, streaming data from multiple sources. Based on open data sources, these technologies are to facilitate faster, more accurate and more complete information delivery for understanding and decision making. This paper is not meant to be a complete review of all relevant publications on the topic, but rather to provide directions for knowledge engineers and researchers to further explore interesting solutions that we believe accord with current trends.

1.1 Knowledge bases

A *Knowledge Base* is generally a repository of domain specific or general knowledge gathered from data sources. Some forms of KB have been used in Expert Systems since the 1970s. In the 1990s, KBs started to be used for storing and reasoning with information extracted from the World Wide Web.

Generally, a KB consists of *entities*, organised in a hierarchical structure from general to more specific, for example fruit > apple, and *relations*, such as 'aka' (also known as), 'father of', 'located in', etc. A KB can, therefore, be viewed as a 'knowledge graph' with entities as nodes and relations as edges. Querying a KB for a specific entity will return a set of related entities together with their relationships. This in turn can be used for question answering or summarisation.

In 2012, Google published a blog about a new type of KB called the *Knowledge Graph*[1], consisting of 'things not strings'. Entity types were extended to include images, videos, landmarks, geo-locations and other entities known to Google. The Google Knowledge Graph allows searching all entities and their relations. In this survey, we use the term *knowledge base* to include such knowledge graphs, as this is the term commonly used in the literature. We refer to 'constructing a KB' to mean the process of finding entities and their relations, and incorporating them into the KB under construction. We do not focus on the underlying technology used to store the KB.

A term related to 'knowledge base' used in both Philosophy and Computer Science is *ontology*. In Philosophy, ontology refers to the nature of existence and the categorical structure of reality. In Computer Science, an ontology is a more specific notion, an organised set of concepts, entities and relations limited to an application domain (Gruber, 1993; Biemann, 2005), for example health care. Such ontologies can be defined at several levels. Upper or foundation ontologies define general concepts, providing a foundation for extension to domain-specific ontologies. The purpose of a domain-specific ontology is to minimise ambiguity and facilitate reuse of knowledge. Although 'ontology' and 'knowledge base' are often used interchangeably in the Computer Science literature, in this paper we use a more restrictive sense of 'ontology' to cover a domain-specific set of concepts and relations, and consider a KB to consist of such an ontology together with a collection of specific entities and relationships between them. Thus, an ontology for a domain is relatively stable, whereas the purpose of KB construction is to 'populate' an evolving KB with a collection of entities and relationships, using a process we call 'knowledge mining'.

1.2 Data mining

Data mining is important for KB construction as these techniques are used, together with methods from natural language processing (NLP), to extract information from data sources to populate a KB. It is useful at this point to distinguish between *knowledge discovery*, *data mining* and *machine learning*, as these three terms are often used without explanation. Knowledge discovery from data (KDD) (Fayyad *et al.*, 1996) is the whole process consisting of data pre-processing (selection, cleansing and incorporation of prior knowledge), data mining and post-processing (querying, visualisation and interpretation of results). Data mining is one step in this process and covers methods used to discover useful patterns and relationships, and may include machine learning, pattern mining and statistical methods. Data mining also has a strong application focus in providing 'business intelligence'. Machine learning emphasises the development and evaluation of algorithms for tasks such as classification, clustering, inference and prediction.

1.3 Knowledge mining

For building useful KBs, it is insufficient to just discover useful information or even incorporate prior knowledge into a data store. KB construction involves the definition of a domain-specific ontology (usually done manually by a knowledge engineer), then the population of the KB by the addition of specific entities and relations (typically using automated methods with user guidance). KB population

[1] `http://googleblog.blogspot.com.au/2012/05/introducing-knowledge-graph-things-not.html`

involves the integration of discovered knowledge into an existing KB and its ontology. Most importantly, as the KB has a logical structure, the incorporation of any new information should preserve consistency.

Two aspects of KB population have been distinguished in recent literature: *entity linking* (or *entity resolution*) and *relation extraction.* Entity linking/resolution is the integration of newly discovered elements of knowledge into an existing ontology, for example that an entity *Barack Obama* is a *person*, a *president*, etc., where the concepts *person* and *president* already exist in the KB. NLP approaches for entity linking include named entity resolution (NER) and named entity disambiguation (NED) techniques. Data mining methods may also be used, such as classification and clustering. *Relation extraction* involves establishing relationships between various entities in an existing KB, for example *Barack Obama—born in—Hawaii*. This type of relationship enables inference over entities to be utilised in further applications (e.g. that Barack Obama was born in the United States). Relation extraction is usually done with NLP techniques supported by data mining.

In this paper, we call the processes used for KB population *knowledge mining*, a term originally used by Nasukawa and Nagano (2001) for analysis of customer reports in a call centre. Knowledge mining has a different sense from data mining in that data is the source from which we mine (as in data mining), whereas in knowledge mining, knowledge is what we aim to discover (as in the mining of natural resources from the earth). It is also possible to consider the mining of knowledge contained in a KB, however that is outside the scope of this article. A variety of methods, both automatic and those involving user feedback, may be used for knowledge mining, but the most interesting and challenging aspect of knowledge mining is the use of the KB itself to improve knowledge mining. Thus, knowledge mining is an ongoing, dynamic, iterative process of extracting and utilising knowledge to further improve the performance of the knowledge mining processes and related applications.

One of the most important changes in data mining in the last decade has been the rise of applications where the volume of data exceeds the capacity of fast main memory to store all the data. This is the core scientific and engineering challenge of so-called 'big data'. Big data continues to provide serious challenges to current machine learning and data mining methods (Furht & Escalante, 2011). However, the rise of big data also provides opportunities to mine these resources with more powerful methods than could be justified before. This is because powerful learning algorithms are prone to overfit and require relatively large samples of data to avoid this problem. Such so-called 'low bias' learners will be essential for knowledge mining over data of low granularity, for example, when learning about very infrequent or anomalous events or entities which may be of significance, without danger of overfitting. The need of processing large volumes of data almost inevitably leads to the requirement to use online versions of machine learning algorithms that avoid the necessity to store all training data in main memory by instead treating the incoming data as a stream. A key issue for such methods is then data dimensionality, specifically, the ability of online learning algorithms to handle data with very high numbers of features efficiently in order to learn predictive models (Agarwal *et al.*, 2014). The issue will be amplified if we want to mine from multiple asynchronous sources to increase the speed and accuracy of acquiring knowledge.

The development of social media in recent years has created additional challenges for building and updating KBs. The first challenge is the already mentioned high volume of unstructured information flow, requiring more reliance on automated processing and keeping the KB up to date. The second challenge is that often texts are short, containing non-standard vocabulary, and are therefore hard to interpret without incorporating a wider context. The third challenge is finding common ground or correlations between entities 'mentioned' in different data sources. This is crucial for building and using KBs, as references in data sources need to be consistently linked with concepts in KBs. The main problem of automatic processing of documents is that machines lack the context to 'understand' texts written by humans, which means that automatically linking references in texts to related concepts in KBs is inherently unreliable. Overcoming this limitation would allow automated extension of existing KBs, creating summaries and answering queries. But for now, much human feedback is required to validate the contents of a KB extracted automatically.

1.4 Structure of the paper

The rest of this article is structured as follows. The next section introduces a general architecture of a knowledge mining system, and describes some existing examples of current work to illustrate the concept

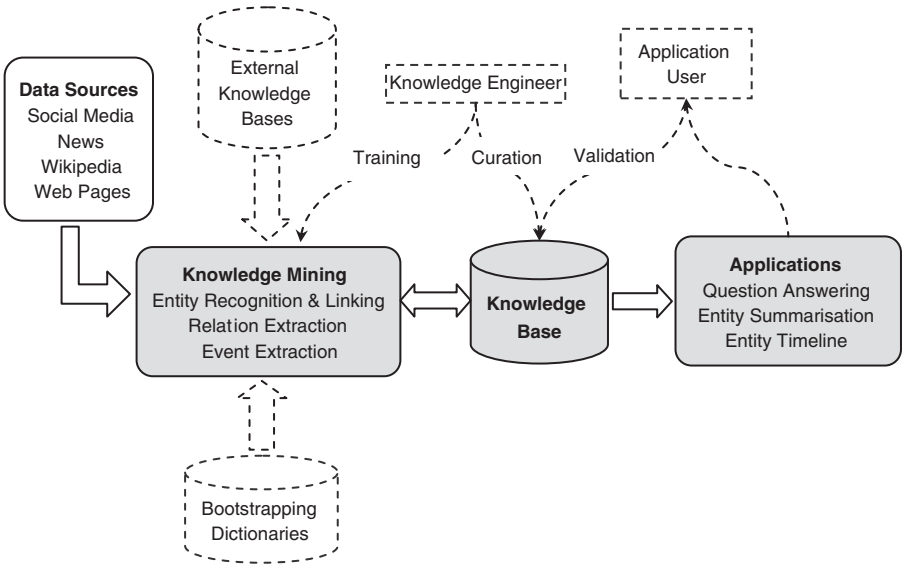

Figure 1 Knowledge mining system architecture

and give structure to the remainder of the survey. We characterise a knowledge mining system as involving knowledge mining techniques, KB construction and applications. Thus, in the subsequent Sections 3, 4 and 5 of this paper, we review the literature related to each of these aspects in greater detail.

In particular, Section 3 discusses the requirements of online data mining environments and covers the elements of knowledge extraction needed to populate KBs. In Section 3.4, we also look at event detection and tracking in data streams, as events can be recorded in KBs as a special kind of entity. Section 4 covers knowledge representations used for structuring KBs, and comparison of various methods for KB construction. Section 5 reviews the typical uses of KBs mainly based on published results of the Text Analysis Conference (TAC) challenges. The final section identifies general trends, indicates potential research areas and concludes the survey.

2 Knowledge mining architectures

A number of knowledge mining systems have been developed that integrate data mining and NLP techniques with KBs, in order to identify relevant sources of information, answer queries, build summaries for documents, events, persons and objects of interest, as well as predict future events. In this section, we propose a general architecture of knowledge mining systems and present three exemplars of such systems with different structure and functionality.

2.1 Typical system architecture

Figure 1 shows the main components of a general architecture for a knowledge mining system. The knowledge mining component is shown as a shaded box with a list of main functions. Its main purpose is mining for KB entries, such as entities, relations and events. These elements are extracted from possibly multiple data sources and matched with the ontology in order to consistently place them in the KB (the arrow from knowledge mining to the knowledge base). These tasks are typically performed by a processing pipeline consisting of text pre-processing, machine learning/data mining methods such as clustering and categorisation, and NLP methods.

NLP methods used for entity extraction include NER and NED techniques. This process is often supported by an external KB and bootstrapping dictionaries, such as a paraphrase dictionary, particularly useful for tweets. The Paraphrase Database (PPDB)[2] is an example of such a dictionary.

[2] http://www.cis.upenn.edu/~ccb/ppdb/

Entity disambiguation can be facilitated by comparing synchronised information from multiple data sources and, importantly, may utilise a partially built KB in an iterative process (the arrow from the knowledge base to knowledge mining). This creates a major challenge for such 'bootstrapping' of knowledge mining systems because for the KB to be used effectively in knowledge mining, the extracted knowledge must be sufficiently reliable. Thus, some form of human input is typically incorporated into the process. Human input typically provides background knowledge in the form of curation rules to improve the quality of the KB, and human-labelled examples to improve supervised machine learning processes. At the application level, user feedback may confirm the credibility of specific pieces of information, which in turn may serve as training data for machine learning algorithms or be used for KB curation.

A significant question is the purpose to which the KB is to be put. If there is no specific purpose, that is, the aim is to extract as much knowledge as possible, the reliability of the knowledge mining process and the consistency and correctness of the resulting KB is a major issue, especially as such KBs soon grow beyond the capacity of people to validate them. On the other hand, if there is a specific intended application, the knowledge mining process is more constrained, so presumably more reliable, however, the accuracy and completeness of the knowledge extracted remains an issue. In general, evaluation of such complex knowledge mining processes and of the resulting KBs is difficult, and mostly absent for the exemplar systems described below. Research on KB applications so far has concentrated on question answering and event extraction, whereas applications to general data mining and text summarisation may become future trends. For these specific applications, there is of course much evaluation of methods on small-scale data sets.

2.2 Existing knowledge mining systems

In this section, we describe representative examples of knowledge mining systems that attempt to capture general human knowledge rather than apply to any specific domain. These systems are to be taken as research prototypes undergoing constant evolution and ongoing development, thus what is most important are the broad approaches, which we show with reference to the general architecture in Figure 1, rather than the specific details, of which some are only sketchily given and thus cannot be described in depth in this article. The diagrams included in this section show our reconstruction of those architectures from the information presented in the cited papers, in order to identify similarities and differences between the approaches.

2.2.1 Kosmix social media analytics

The Kosmix social media analytics platform (Chai *et al.*, 2013) is an example of a typical system architecture as depicted in Figure 1 for knowledge mining and usage, with a KB, usually called Kosmix KB, as a central component. Data from a variety of sources, such as Wikipedia, Foursquare, Chrome and Twitter, are processed to extract and disambiguate entities and detect events. Tweets are ingested and processed in near real time, mostly for event detection. For this purpose, a highly scalable infrastructure has been built, consisting of a file system, relational database management system (RDBMS), Hadoop and Muppet (a map-reduce style distributed system with update function[3]). These processes are supported by crowdsourcing, consisting of internal analysts, Mechanical Turk workers and users, in order to increase the accuracy of extracted information.

A basic KB is initially constructed from Wikipedia and additional sources, such as Chrome and Yahoo. A top taxonomy of categories is created using a mixture of automated methods and human input, for example recommending a parent to a node. The KB is re-built afresh every day and curated by human-generated rules. Thus, a lot of effort is dedicated to updating the KB, especially reapplying the curation rules after each update run. On top of the basic KB, additional information is added: profiles and tweets from Twitter, events from the Twittersphere and raw data from web pages. In order to augment the

[3] Muppet has been released as open-source software under the name of Mupd8.

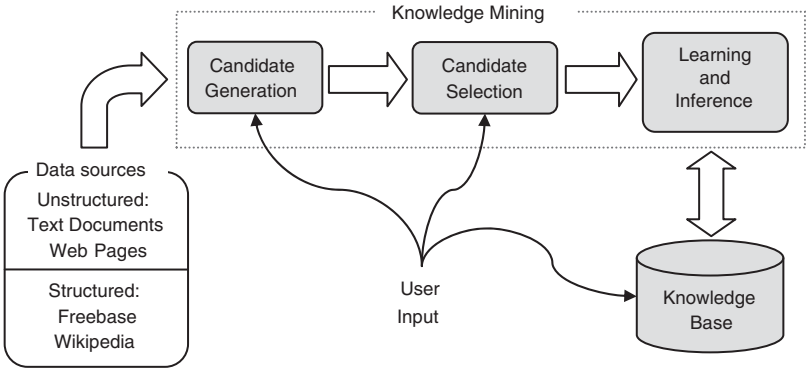

Figure 2 DeepDive system architecture

basic KB with social media data, entity extraction, linking, classification and tagging are performed applying context and social information, and using Kosmix KB, in a synergetic manner.

Kosmix KB has been used in a number of applications, including TweetBeat (detection of emerging events), Firsthand (a news assistant) and SocialCube (question answering with location, topic and sentiment). A number of conclusions have been drawn from building and using the system (Gattani *et al.*, 2013). First, it was difficult to accurately extract entities with some generic names, such as 'Thank God It's Friday' as a title. Capturing context is critical for this task. Second, even a KB with some inaccuracies is useful in many applications. By using human curation, the accuracy of the KB taxonomy lifted from 70 to 90% (Deshpande *et al.*, 2013).

2.2.2 Stanford DeepDive

DeepDive (Shin *et al.*, 2015) is an open-source system for KB construction. It takes a variety of data sources: unstructured (text documents), semi-structured (Wikipedia) and structured (Freebase) and creates KB entries. Figure 2 identifies the knowledge mining stage from Figure 1, with the KB as the final product. Results from all stages of processing are stored in an RDBMS and most processes in DeepDive are based on SQL statements. The knowledge mining stage consists of a number of steps. In the first step, a combination of SQL queries, NLP methods and user-defined functions perform feature extraction, producing sets of candidate entities, relations and 'mentions'. Candidates are selected in the second step by classifying each element as true or false using a combination of hand labelling and distant supervision. In the third step of knowledge mining, denoted 'learning and inference', a factor graph is created that connects evidence with candidate facts. In this step, statistical learning and inference is used to resolve inconsistencies between candidate facts and to produce the marginal probability associated with each KB entry. For example, the relation HasSpouse(Barack Obama, Michelle Obama) would be expected to have high probability. Human assessors are involved in this stage to learn common errors and correct them, again using SQL statements, before entries are accepted into the KB.

Scalable relational technology allows avoiding bottlenecks, especially in the grounding and statistical inference stage (Ré *et al.*, 2014). The use of RBDMS allows applying incremental view maintenance, a technique known in database technology (Gupta *et al.*, 1993) for incrementally modifying the view with the change of underlying input data. The main difference with the Kosmix system is that DeepDive does not create a general purpose KB, but can be used for domain-specific KBs, such as geology and pharmacology. Therefore Figure 2 does not show the applications box.

Examples provided in Shin *et al.* (2015) suggest that DeepDive works better with richer and structured documents, as opposed to loosely structured tweets. According to the DeepDive website, the system achieved 'winning performance in entity relation extraction competitions', referring to the TAC 2014 slot filling competition (Angeli *et al.*, 2014). The comparison to other systems in that competition, however, is not published. We noted, however, based on the above system description, that much human intervention is required to produce these results for the competition.

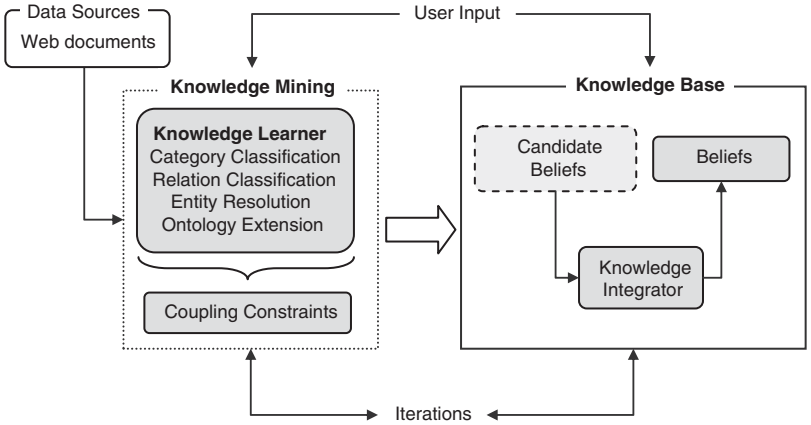

Figure 3 Never-Ending Language Learning architecture

2.2.3 CMU Never-Ending Language Learning

Never-Ending Language Learning (NELL)[4] (Mitchell *et al.*, 2015) is a semantic learning system that autonomously reads information from web pages and incrementally builds a general KB. At a high level (Figure 3), NELL performs two main tasks: (1) extracting information from the Web to generate candidate entries for the KB (what we call knowledge mining), and (2) using machine learning and human input to improve and correct KB entries (knowledge base with user input and iteration loops). This structure is our reconstruction of the architecture presented in various papers, roughly conforming to the diagram in Figure 1.

The knowledge mining part of the system contains the knowledge learner, running a number of extraction tasks. Figure 3 lists only the categories of these tasks under the knowledge learner label. Each category contains many such tasks, each dedicated to a specific context, feature or extraction method. These numerous tasks have some mutual dependencies, hence the coupling constraints module within the knowledge miner executes constraints and monitors their satisfaction. The extraction task extracts two types of information: entities for each pre-defined category of entities and a pair of entities for each pre-defined category of relationship. The ontology extension module finds new relations for entities already in the KB.

The NELL KB architecture depends fundamentally on modelling belief with provenance (source path) information and the use of confidence levels. A key objective is that an accurate and consistent KB can be constructed from large amounts of data with relatively little human input. To achieve this, KB population and learning is designed as an iterative process, using an approach similar to expectation maximisation in the following way. Knowledge mining produces candidate beliefs that can be promoted to full KB beliefs by the knowledge integrator, which also processes deletion requests to particular beliefs. In the next iteration, the partially built KB is used to retrain the models produced by the knowledge learner.

Given the lack of labelled data relative to the massive amounts unlabelled data on the Web, NELL relies on a paradigm called ‘coupled learning’. Without going into detail, the key idea is that although individual functions may be hard to learn separately, by learning thousands of functions together the problem becomes more constrained and hence easier. This generalises semi-supervised learning (Section 3.1), and relies on coupling constraints (Figure 3) which can be user supplied or learned. A further key aspect is that learning and KB expansion happens in a staged fashion as more data are gathered and the KB and ontologies grow, thus providing more constraints for subsequent learning iterations.

Two kinds of human input are provided to NELL: (1) human-labelled training examples for each entity and relation category, and (2) user feedback through NELL’s website, where users can vote each belief up or down. These corrections are turned into further training examples (although it is not clear exactly how

[4] `http://www.cmu.edu/homepage/computing/2010/fall/nell-computer-that-learns.shtml`

this is done). It seems that this feedback contributes significantly to improving the accuracy of system knowledge. Empirical evaluation of NELL has been limited: some papers confirm an improvement in accuracy of the KB over time, however, we are not aware of any publication evaluating NELL in an external application. This highlights a serious problem, not confined to NELL, which is that such a highly complex and mainly automated process involving numerous tightly coupled modules is in some sense an autonomous discovery system, and hence hard to evaluate objectively. Thus, analysing the knowledge mining process and evaluating any improvements in the accuracy of the resulting KB is extremely difficult, at least given the current state of the project.

3 Knowledge mining for knowledge bases

In this section, we provide an overview of data mining techniques for building KBs, focussing on stream mining methods combined with NLP, then examine techniques specifically for entity extraction, relation extraction, and event detection and tracking.

3.1 Overview of data mining techniques

There are numerous data mining techniques currently in use which are covered in a number of introductory texts, such as Han *et al.* (2011) and Witten *et al.* (2011). Two main categories of data mining are clustering and categorisation. Clustering is useful in topic and trend mining, while categorisation can be used for extracting and classifying entities and relations into relevant KB categories. Some methods, such as *k*-Nearest Neighbour and Naïve Bayes, are very efficient in online stream mining, whereas others, for example support vector machine (SVM), work better in offline settings. Decision tree based categorisation techniques combined with probabilistic methods are worth mentioning as they are very efficient and facilitate decision transparency (a way to explain the classification decision). Well-known methods are VFDT (Very Fast Decision Tree, based on the popular C4.5 algorithm) and Concept-adapting Very Fast Decision Tree learner, a VFDT extension to handle streaming data (Hulten *et al.*, 2001).

Supervised classification algorithms require correctly labelled examples, which are often difficult to obtain. Some methods give acceptable results with only partial labellings or in a semi-supervised mode (Carlson *et al.*, 2010; Liu *et al.*, 2011, 2013). Semi-supervised learning relies on very large collections of web documents and information redundancy. This is based on combining a small amount of data on known concepts or relations with a much larger amount of data where this information is unknown. The known information is then used to seed predictions, which are reinforced if multiple independent sources of information can be used to make the same prediction with a learning algorithm. In this way, confidence in that prediction should be increased. Another option to overcome the scarcity of annotated training examples is the use of external or built-in-progress KBs. This method, called distant supervision (Mintz *et al.*, 2009; Hoffmann *et al.*, 2011), has been used for example by Roth *et al.* (2013) for predicting relations in the TAC competition.

Another common problem in text mining, relevant to entity matching in KB construction, is the selection of methods for comparing texts. A wide range of methods are used (Han *et al.*, 2011), starting from Jaccard and cosine similarity for a 'bag-of-words' approach, where keywords are treated as vector features, disregarding their semantic relationship. Some semantic features are captured in the *n*-gram approach, where word counts are made over windows of *n* keywords for a large number of documents and the resulting counts used as features for comparing documents. Probabilistic Soft Logic (Bröcheler *et al.*, 2010) can be used for a deeper semantic similarity calculation (Beltagy *et al.*, 2014), combining logical and distributional information. Generally, better results are obtained when using semantic features (Ji *et al.*, 2010; Ji & Grishman, 2011; Guo *et al.*, 2013; Beltagy *et al.*, 2014; Gao *et al.*, 2014).

The use of rules has been recognised as providing effectiveness and transparency. In particular, ripple-down rules (RDR) (Compton & Jansen, 1990), an incremental knowledge acquisition method capable of handling large numbers of rules, has been applied to text classification tasks for news and e-mail (Ho *et al.*, 2003; Park *et al.*, 2004; Wobcke *et al.*, 2008) and information extraction (Pham & Hoffmann, 2005; Kim & Compton, 2012a). These methods can be extended to the above-mentioned text matching

problem and for KB construction. Suganthan *et al.* (2015) similarly use hand-crafted rules for product classification in a recently developed system called Chimera used at WalmartLabs. They argue that, in practice, it is often both necessary for achieving sufficiently high accuracy, and also much easier, to define classification rules than to capture relevant knowledge automatically, for example 'if a product description has an ISBN then it is a book'. Part of the Chimera system is the generation of new rules from labelled data using a variation of the Apriori Algorithm (Agrawal & Srikant, 1995): the rules are then validated by system designers. Rules are also extensively used for KB maintenance and curation to correct gaps in automatic knowledge mining, for example in Kosmix (Deshpande *et al.*, 2013).

In our view, the high volume of data makes stream mining (Gama, 2012) more suitable than collecting, storing and analysing data in a batch mode fashion. By a *data stream*, we mean a (theoretically infinite) temporal sequence of data, which can be numerical data (such as stock prices) or multimedia documents (including text, audio, image and video, or a mixture of them) as seen from a particular point of reference, for example tweets can be viewed from many points of reference: topic, sender, recipient, etc.

Streaming sources are characterised by an abundance of data whose properties *change over time*. Therefore, stream mining techniques generally have a lower accuracy than batch methods (where all data are available in advance) because typically only a small number of data items can be stored, and items in a stream are processed one by one in sequence. To some degree of compensation, stream mining algorithms may have a large number of training examples and may exploit information implicit in the temporal ordering of the stream to improve accuracy. Thus, stream mining algorithms attempt to strike the right balance between speed, accuracy and the computing resources required to store, process and derive conclusions from data.

As the data stream is virtually infinite, the data mining model usually changes over time. This change, called *concept drift*, can be detected and incorporated into a model if needed (Gama *et al.*, 2014). Commonly, such methods rely on detection of concept drift (Baena-García *et al.*, 2004; Gama *et al.*, 2004) or adaptation to change by gradual forgetting (Koychev, 2000) or windowing (Widmer, 1997). Another adaptation approach is to maximally utilise periods of time when the concept is stable, called *concept clumping* in our previous work (Krzywicki & Wobcke, 2010, 2011).

When dealing with fast streams, windowing and automatic adaptation techniques (e.g. Bifet & Gavaldà, 2006), are more useful than sampling as the latter may miss the detection of important events, for example short bursts of tweets related to a particular topic. This is especially important in security applications, where infrequent and anomalous patterns need to be detected.

3.2 Entity extraction

Entity extraction refers to finding 'entities', such as people, places and dates, in data sources. This is a non-trivial task as, apart from standard entity tagging, which can be performed by NLP software, there is also a need to establish which of the different entities mentioned in different streams or documents are, in fact, the same entity (the problem of *entity resolution*). Entity resolution cannot usually be done with NLP techniques alone, and requires more specialised algorithms, often based on dictionaries or KBs. Even the seemingly simple task of entity tagging can be difficult for some data sources, such as Twitter, where non-standard vocabulary is used and messages often form ill-structured sequences. For example, the word 'tomorrow' is used in over 50 different lexical variations (Ritter *et al.*, 2011). These difficulties are commonly alleviated by using dictionaries, gazetteers and KBs.

The results of entity resolution need to be ranked based on the probability of matching. This often leads to the situation where entity tagging, extraction and resolution tasks are hard to separate and are built as an iterative process. Some examples of research reviewed in this section support this view. One such example is the research of Davis *et al.* (2012), which gives a fast, incremental algorithm based on the expectation maximisation framework. The lazy association classifier (Veloso *et al.*, 2006) is used to perform fast, incremental learning. Starting to learn from a few positive examples for each entity, the algorithm turns some negative into positive labels (or vice versa) in the validation phase. The classifier is updated partially after each label transition and is able to determine the set of rules that need to be updated.

The process is repeated approximating the true positive–negative distribution. The method is more accurate and much faster when compared with retrained SVM (Cortes & Vapnik, 1995).

The above approach involves shallow learning from large amounts of data and is typical of many streaming algorithms. Another fast, but much more complex method, designed for the Kosmix system to work with the Twitter firehose, is presented by Gattani *et al.* (2013). This method combines entity extraction, linking it to a Wikipedia KB entry, classification into a pre-defined topic and assigning KB concept names as tags to tweets. Entities are not only people and locations, but also products, music albums, songs, books, TV shows, sports events, cars and movies. The whole process integrates all these steps together and uses web context for tweets and social context for users to enrich information in tweets. Mentions of entities are scored by their similarity to the contents of links from tweets, tweet tag and KB. Filtering, scoring and tagging is done again as more features are extracted about mentions. Finally, editorial rules are applied to improve accuracy. The method was evaluated using a small ground truth sample of 99 tweets processed manually, and results were compared with Stanford NER and OpenCalais.

Liu *et al.* (2011) use *k*-Nearest Neighbourhood (*k*-NN) to pre-label words across tweets, and then conditional random fields (CRF) (Lafferty *et al.*, 2001), is applied for fine-grained, tweet-level NER. CRF uses extra information from gazetteers covering common names, countries, locations, temporal expressions, etc. Here gazetteers play the role of a very simple KB. The *k*-NN and CRF models are retrained after a pre-defined number of tweets are processed. For evaluation, 12 000 manually annotated tweets are used, and precision, recall and F1 (harmonic mean of precision and recall) metrics used as performance measures. Results were compared with two baseline methods: gazetteers dictionary lookup and the target system, but without using *k*-NN and semi-supervised learning. Recall and F1 were higher on all types of tested entities, while the precision was lower than the baselines. The algorithm can process tweets in a stream-like fashion, but the execution time is not provided.

While the above methods are geared towards fast, online processing, it is possible to perform entity resolution offline as long as the process can be updated and fast enough to catch up with new entities arriving from the stream. The next two methods belong to this category. Both of them use a version of Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003) for labelling, and a KB to constrain the disambiguation. Ritter *et al.* (2011) propose an NER method comprising a number of components: part of speech tagging, parsing and NER. Clustering and CRF are used to perform these tasks. A separate SVM model is used to determine if word capitalisation in tweets is informative. Freebase is used for 70% of entities, and the other 30% are labelled using Labeled LDA. The evaluation section reports an improvement over Stanford NER using manually annotated tweets. Most of the components were trained using annotated tweets, which is not a sustainable solution for continuous data streams with concept drift.

Li *et al.* (2013) developed a method for Twitter entity resolution by supplementing KBs with extra evidence from internal documents in the KB and external corpora. Related names to a given entity are first extracted from the KB and then relevant documents are searched for the extra evidence. These other documents are labelled using LDA with labels related to the entity in question and are used to disambiguate the entity. The process of gathering extra evidence can be run offline and used by any disambiguation algorithm. Results compare favourably with Wikifier (mapping entities from given text to Wikipedia) and AIDA (an NED system making use of a weighted mention-entity graph to find the best joint mention-entity mapping). We note, however, that these two methods were not designed or tested on Twitter text by their original authors, therefore the comparison is less reliable.

Although Twitter entity resolution presents a main challenge, there are a number of methods designed to work with other types of documents, such as news and blogs. Li *et al.* (2011) proposed person identity matching in databases, based on both personal features (name, date of birth, social security number, etc.) and social features (participation records in social events or crimes, and their roles in these events, e.g. suspect, victim). For matching personal features in the form of strings, a normalised Levenshtein edit distance (Levenshtein, 1966) was used, which is the number of edits, deletions and substitutions to transform one string into the other, divided by the longer string size. Social features are compared using a number of similarity measures, such as Jaccard coefficient. Similarity measures are weighted for the contribution to the final similarity decision and weights are learned using Naïve Bayes. The evaluation data set was real police records and the gold standard was created for individuals with the same last name

and date of birth. Pre-clustering was used to avoid comparing all pairs. However, adding social features improved the accuracy of the method by only about a couple of per cent.

In Kotov *et al.* (2011), named entities are extracted from correlated bursts of mentions in multiple streams of news articles. The process consists of two stages. In the first stage, bursts of entity mentions are discovered in streams and modelled with a Markov-modulated Poisson process (MMPP) to estimate the temporal coefficients of bursts. In the second stage, these coefficients are used in a dynamic programming algorithm to discover all pairs of streams with temporally correlated bursts. Apart from correlating entities, the process is able to transliterate entity names between different languages. The baseline chosen for comparison is the binning-based normalisation method, which is a simplified MMPP. Results show higher recall values of the method described over the baseline.

An interesting method for NER is presented by Kim and Compton (2012b), where Stanford NER is combined with RDR to deal with the informality of web documents such as tweets. The approach is not intended as an alternative to Stanford NER or other general entity recognisers, but to improve their performance in a specific domain. The RDR KB is built incrementally while the system is in use; therefore it is able to update the KB as needed. This paper shows examples of errors commonly made by Stanford NER and corrected with RDR and presents a comparison of RDR with Stanford NER on the MIL (Multiple Instance Learning) data set, which is a sample of sentences typically used on the Web. On all types of entities, RDR is able to improve the performance of Stanford NER. Additionally, it requires a smaller number of documents compared with other NER methods based on machine learning.

Geo-locations are important features, for example in security and marketing applications, but if not provided directly, identifying them in streaming data is challenging and requires contextual understanding of messages. Often multiple locations are reported, such as the location of the incident and the location of the reporter. In the EMBERS system (Ramakrishnan *et al.*, 2014) location (city, state and country) is detected using Probabilistic Soft Logic (Bröcheler *et al.*, 2010) with a model that combines and weighs different evidence types, such as location tokens, and organisation and person names. In tweets, geo-location features are sparse and an explicit location is rarely given. Cheng *et al.* (2010) developed a method for identifying a city-level location for tweets, even in the absence of location tokens. The method is based on the observation that specific keywords have high local focus, and when taken together, can provide a probabilistic identification of a place. The challenge is the selection of appropriate keywords. For this purpose a set of keywords was analysed and selected with satisfactory geo focus and dispersion. In testing, the method was able to identify 30% of users within 10 miles. This may not be sufficient for many applications, but it gives an idea of the difficulty of the problem. Better results are obtained by Li *et al.* (2011), who explore two kinds of relationships: the probability of following a given distance and the probability of using venue names given the user distance to the venue. The method places around 50% of users within 10 miles.

In summary, NER and NED methods for entity extraction rely on a KB or multiple streams of documents. Common problems found in the methods are the need for prior training on fixed, annotated documents, which hinders their application in streams, and poor data set and baseline selection for evaluation and comparison.

3.3 Relation extraction

The focus of this section is on extracting relations that connect entities in a KB. Zhu *et al.* (2009) extend the Snowball method of Agichtein and Gravano (2000) to start with a handful of examples (bootstrapping) and iteratively generate new extraction patterns together with their weights using a Markov logic network (MLN). Patterns are selected using ℓ1-norm regularised maximum likelihood estimation. Their KB system called StatSnowball can extract pre-defined relations as in Snowball or a general type of relation. General patterns are built using shallow NLP: at the entity level these patterns are part of speech tag sequences between entities selected using the MLN. At the sentence level, relations are extracted taking into account tokens surrounding the entities. Evaluation shows that StatSnowball performs better than CRF on all tested entity types except verbs.

Bootstrapping is a popular technique to find high probability relations as seeds and then use them as training examples to find more relations. Bootstrapping dictionaries can be built manually or learned,

starting with a handful of seeding examples for a semantic category (Riloff & Jones, 1999). Etzioni *et al.* (2005) start with automatically created general extraction patterns from well-known relations, for example 'such as', then create extraction rules that help to extract more patterns. A common problem with automatic extraction, however, is semantic drift, where the meaning of subsequently extracted patterns drifts from the original seeds. Curran *et al.* (2007) present a method to reduce drift using mutual exclusion bootstrapping. Entities are divided into a number of semantic classes under the assumption that they are mutually exclusive (do not contain the same entity names) to prevent entities from crossing class boundaries. Despite the automated extraction process, the method requires a substantial amount of human knowledge (semantic classes, stop classes, filtering rules). Another drawback is that all *n*-grams need to be in memory at processing time. In summary, bootstrapping starts with well-known relations, using them as training examples, then proceeds to extract more relations while controlling semantic drift.

The Kosmix KB (Deshpande *et al.*, 2013) is constructed from Wikipedia and other data sources. Relations for the KB are extracted in a variety of ways from Wikipedia info boxes, templates and the article text. Each source of relations is associated with certain relation categories by extraction rules. A small number of extraction rules can be used to extract a large number of relationships. Synonyms are captured from redirection pages and homonyms from disambiguation pages as well as additional properties, such as Twitter id, Web URL and co-occurring concepts.

Roth *et al.* (2013) present the TAC 2013 winning algorithm in the slot filling competition. The task is to find relations for entities given in queries. Query expansion, entity retrieval and candidate matching are required as first steps. Each type of relation is modelled by training a distantly supervised SVM classifier. The distant supervision training set is obtained from Freebase relations and from hand-crafted seed patterns. Prediction is done for each pair of entities and a candidate sentence which could contain the relation.

An important conclusion to draw from the above review is that extracting entities and relations cannot be done using one well-defined method and rather requires an integrated approach guided by human judgement.

3.4 Event detection and tracking

Event detection and tracking in data streams is important in many domains, such as national security, for a number of reasons. First, predicting events, such as civil unrest, allows for resource planning and deployment for preventive action. Second, detected events can be linked with particular individuals and places, which helps in building a summary of interesting events related to an entity or location.

Before reviewing state-of-the-art work in this area, we first look at the definitions of events, and in particular, how to distinguish events from non-events. It is remarkable that, despite a rich literature in Computer Science on event detection, the notion of an event is rarely defined and often the definition can only be inferred from the method description. This is the case for Schrodt *et al.* (1994), which is the earliest work on automatic event detection. A sentence defines an event if the sentence structure is subject–verb–object and the verb is in a dictionary of pre-defined event verbs (thus a class of events is defined by stipulation). In research by Huang and Riloff (2013), event recognition is also based on a dictionary, which serves as a bootstrap for learning event agents and phrases. Fung *et al.* (2005) defines an event as a minimal set of 'bursty' features occurring in a time window over a maximal set of documents (which clearly could also cover non-events). SARS[5] is given as an example of an event, however, the main problem with this definition is that bursty features may refer to both events and topics. One could argue that SARS, the disease, is a topic rather than event. Becker *et al.* (2011) define an event in the context of Twitter as an occurrence associated with a time period and discussed in a set of messages (which again seems too general). Yang *et al.* (1999) list document features specific to an event: time proximity, bursty numbers of documents, change in vocabulary and a relatively brief time window of occurrence.

While the above Computer Science definitions are focussed on how events are identified, the psychological approach to events takes a different perspective (Zacks & Tversky, 2001). According to this

[5] Severe Acute Respiratory Syndrome, a viral disease with flu-like symptoms, an outbreak of which occurred in southern China between November 2002 and July 2003, resulting in numerous deaths.

definition, events are perceived as objects bounded in space and time. For example, SARS is an event as long as associated messages talk about a particular SARS outbreak in a particular place, and are not about treating SARS, etc. The main difference between events and topics is that, unlike objects, single particular events can be experienced only once.

In our view, the basic property of events which distinguishes an event from a topic is that an event involves a change of state, that is states are conceptually prior to events, and events are defined by state changes. In the above example, a SARS outbreak in a particular place and time causes a change in the number of sick people, etc. From a linguistic perspective, Monahan and Brunson (2014) recently discuss a number of such qualities of 'eventiveness' of event descriptions: occurrence (signalling a change in state), spatio-temporal grounding, lexical aspect (time boundedness and duration), agency (degree of actor control over an event), affectedness (degree of affect on event participants), actuality (modality of an event description, relating to possibility or uncertainty) and specificity (generic, habitual or specific). The idea of eventiveness may be useful in the context of event mining, insofar as this involves the analysis of documents containing event descriptions.

As with other applications of data stream mining, detecting events in streams presents additional problems in terms of memory and stream catch up time. Yang *et al.* (1999) extract events from news documents using two clustering algorithms: group-average multi-pass hierarchical (GAC) and single-pass non-hierarchical clustering (INCR). Terms are weighted by *tf-idf* (term frequency, inverse document frequency) and the distance measure is standard cosine similarity. The INCR method simply clusters documents using a pre-defined distance threshold and uses a linear decay factor in a fixed window for down-weighting older documents. Event tracking is treated as a separate fast learning task for which *k*-Nearest Neighbour and Decision Tree classifiers were selected. A non-C4.5 Decision Tree was specially constructed for fast incremental learning and achieved the best results in event tracking as measured by the micro-F1 metric. For the event detection task, as expected, GAC achieved better results, but we believe with worse time efficiency, not reported in this paper. Another problem is the sensitivity of the INCR method to the threshold, which needs to be selected manually. A threshold that is too small may pick up noisy events, while one too large may cluster a number of different events together.

The work of Fung *et al.* (2005) mentioned above detects events based on the probability of features in a binomial distribution. Then the minimal number of bursty features is determined to make a bursty event by maximising the probability of the set of bursty features in a document stream. Experiments conducted on 66 300 news documents confirmed that events can be detected, but quantitative measures are not provided, that is precision, recall, F1, etc. The algorithm is an offline processing method, although an incremental extension may be possible by running bursty feature identification and event detection in parallel.

Becker *et al.* (2011) attempt to construct an online method for event detection in the Twitter stream with the main goal of distinguishing events from non-events. The procedure adopted is to first cluster tweets and then eliminate non-event clusters. The clustering method is similar to INCR (Yang *et al.*, 1999), with the threshold parameter tuned during the training phase. Weka classifiers (Witten *et al.*, 2011) were used for determining event clusters trained on a set of temporal, social, topical and Twitter-centric features. Evaluation was performed on 374 annotated clusters used for training and 300 clusters for testing. Results measured as F1 are only compared with the standard Naïve Bayes algorithm showing the superiority of their approach. Despite using an online clustering technique, very simple in fact, nothing else suggests that the whole process can be used for real stream event detection and it was not evaluated as such. It is not known how the threshold was set during the training phase. Use of a fixed threshold would certainly not work as well in a data stream scenario with concept drift. The authors admitted that this problem is not the focus of this paper but the subject of future research. The second problem for real online implementation is the two-stage clustering and classification process. This can possibly be resolved as two independent processes running in parallel with some implications on stream catch up time.

Becker *et al.* (2012) focus on retrieving documents from social media about an event identified by title, description, date/time and venue in typical event sites, such as Last.fm, Eventbrite and Facebook events. The process consists of two steps: (1) a precision-oriented query to first retrieve, perhaps a small number, of documents highly relevant to the event, and (2) a recall-oriented step to expand the query using terms from the first step. The method was trained on 329 event records and tested on 393 events. In the precision

step, documents related to the event were retrieved and scored by the number of queries returning a particular document. In the recall step, *n*-gram joint probabilities were used, returned by the Microsoft Bing search engine. Queries were generated and ranked using a number of *n*-gram and temporal techniques. Queries and retrieved documents for a random set of 60 events were manually annotated and compared with machine results using the Jaccard coefficient. The most similar results to annotated documents were obtained from Twitter using the Microsoft *n*-gram score. Interestingly, combining Twitter documents with Flickr and YouTube tags did not improve overall results, indicating a low accuracy of tags. Although the whole method described in this paper is offline in nature, it may be interesting from the point of view of enriching online information for use in constructing event summaries.

The research of Huang and Riloff (2013) introduced above uses a seeding dictionary as a bootstrapping technique to recursively harvest more event expressions and identify events. Results obtained on a small sample are successfully compared with SVM. The method, although interesting, is not suitable for online implementation as it requires multiple scans of documents. Silva and Riloff (2014) found that pre-classifying users into user types helps with event recognition.

Finally, in specific domains, users may not be interested in all events but only in some class of events. For example, a football match may not normally be an interesting event, but may be of interest if a threat is involved. Interestingness measures (McGarry, 2005; Geng & Hamilton, 2006) may need to be developed or extended for events in further research, and potentially applied to all the above methods. In addition, most of the methods discussed above are implemented and tested as offline processing systems; therefore additional research is required to adapt them to data stream processing.

4 Knowledge base construction

In this section, we discuss specific issues related to populating, updating, curating and using KBs, especially those built from data streams, and consider the important question of evaluating the quality of the resulting KBs. We start with a brief summary of approaches to representing KB ontologies, to make the type of KB under consideration in this article more concrete.

4.1 Knowledge representation for ontologies

A simple knowledge representation scheme at its most basic layer are triples <subject, predicate, object>, for example <'Tom', 'has son', 'Charlie'>. The next layer expresses types and classes of concepts and relations and is represented by an ontology language. The most popular languages for representing ontologies are Resource Description Framework (RDF) (Pan, 2009) and Web Ontology Language (OWL) (Antoniou & van Harmelen, 2009), both based on XML. These languages provide common portable formats for exporting, storing and importing ontologies in files. RDF is the W3C recommendation for semantic annotations in the Semantic Web and provides a formal description of the <subject, predicate, object> triples in the form of XML statements. It is limited to a class and property hierarchy with domain and range definitions for these properties. OWL, in contrast, allows for defining cardinalities of object relations, data-type attributes and the use of logical operators in definitions, for example union of classes. The next level of representing ontologies can be seen as graphs and patterns and are more related and closer to actual sources of knowledge. Some examples of these representations follow.

The Snowball system (Agichtein & Gravano, 2000) extracts relational tuples from newspaper documents and stores them in database tables for using in querying and data mining tasks. Tuples are extracted using patterns of pre-defined relations, such as <LOCATION> -based <ORGANISATION>. This requires a small set of seed patterns and document collection to generate high coverage patterns. Similarly, the Knowledge Vault (Dong *et al.*, 2014) stores information in the form of RDF triples (subject, predicate, object), for example (<person>, born in, <place>).

Kumar *et al.* (1999) represent an early approach to extracting knowledge from the Web (spread sources of information). Knowledge was represented as a web graph. The technique they used was to extract information from subgraphs based on topic-related web pages. This model identifies cliques as sets of web pages containing links to one another. By measuring links between them and applying filters on these

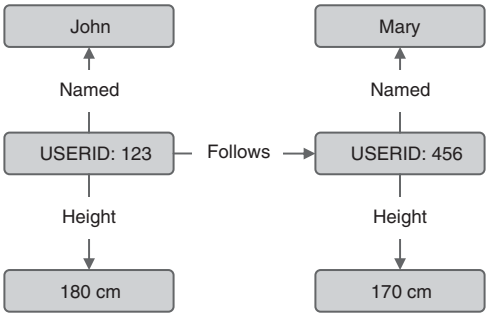

Figure 4 Cayley graph structure

metrics, it was possible to identify and index communities on the Web and create a KB of web communities.

A different technique is used in the Cayley knowledge graph[6], a successor of the Google Knowledge Graph. Instead of storing all entity properties in a database table, each property is represented as a relation (Figure 4), for example person ⇒ height. At first this seems an excessively complex way of representation, but has the advantage of flexibility in accessing, changing and updating the graph.

4.2 Building, maintaining and updating ontologies

Building, maintaining and updating ontologies generally refers to a two-stage process: (1) finding entities and their relations, and (2) matching them with existing entries in an ontology to expand or update its contents. The generation of ontologies has been extensively investigated and implemented with a number of different techniques. Figure 5 shows a basic classification of methods.

The first group of methods to build ontologies are manually based, with Freebase (Bollacker *et al.*, 2007) and DBpedia (Mendes *et al.*, 2012) being the most representative examples. In Freebase, the most remarkable achievement is scalability and collaboration. The platform allows people around the world to collaborate in the construction of the ontology. The ontology is stored in a graph-based structure to allow easy storing and updating of entities and its relations. DBpedia is another example of the crowdsourcing generation of ontology. The main difference is that the former works with the English language, the latter is being constructed from several languages.

On the other hand, automated methods have been recently designed to tackle the main problems of a purely manually maintained ontology. There are different approaches to automated and semi-automated generation of ontologies. Bottom-up techniques address the generation problem from instances, collapsing entities into related groups or categories. These are represented as hierarchical models (e.g. Van Dyke Parunak *et al.*, 2007). Top-down methodologies begin from a general concept and spread the conceptualisation to more specific concepts. XTREEM and TRUCKS are examples of such techniques (Brunzel, 2008; Maynard *et al.*, 2008).

Unsupervised learners, such as KnowitALL (Etzioni *et al.*, 2005), are based on pattern recognition extraction rules that identify classes and subclasses across the Web. An unsupervised mechanism allows automating the lengthy manual process of generating the basic categories of knowledge in a structured way. The main problem of unsupervised techniques is managing ambiguity. The Kosmix KB (Deshpande *et al.*, 2013) is constructed by processing a Wikipedia XML dump, treating pages as nodes and links as edges. This does not create a taxonomy automatically, as there are cyclic dependencies to be removed and top categories to be fixed. The top categories are pre-defined and placed below the root of the tree. The main problem is that automated generation of a KB is noisy, resulting in inaccurate 'facts' and rules that have to be curated and checked for consistency.

[6] `http://google-opensource.blogspot.com.au/2014/06/cayley-graphs-in-go.html`

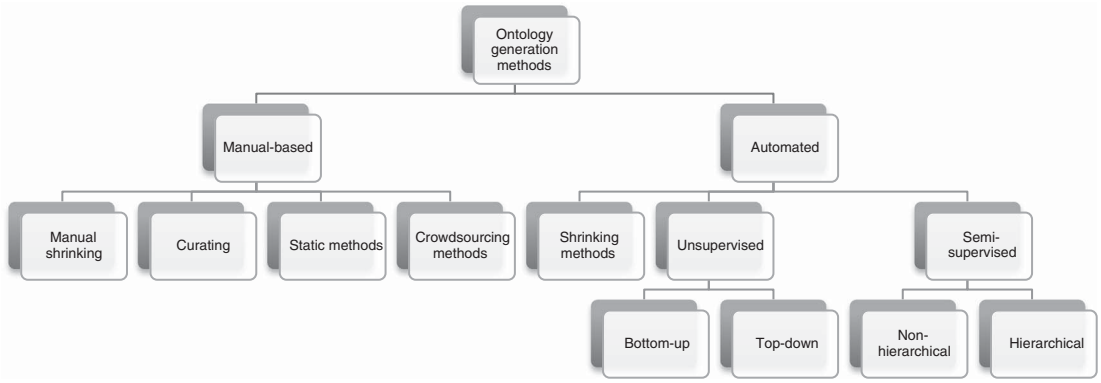

Figure 5 Classification of methods for ontology building and maintenance

Semi-supervised methods such as Dynamo (Ottens *et al.*, 2007) and expert-based ontology generation like Protégé (Tudorache *et al.*, 2008) face the high cost of manual disambiguation using expert knowledge when curating and maintaining the KB. The main problem for supervised and semi-supervised methods is the lack of training data. Some large annotated data sets, such as Gigaword (Napoles *et al.*, 2012), Linguistic Data Consortium data sets or the Multi-Perspective Question Answering corpora, may be available for training.

Ontology generation from scratch and ontology update with new entries may use similar methods, though with different considerations. Large KBs require two intervening steps: regular updates (re-building from scratch or incrementally) and maintenance (curation and consistency checking). These two tasks may interfere with each other and different approaches are needed to make them work together. In the Kosmix KB (Deshpande *et al.*, 2013), the solution is to capture most of the human curation rules as commands, then apply these commands again when the KB is refreshed, once each day. Refreshing the KB is done by re-running the entire KB generation method from scratch. An alternative is to maintain the consistency of the ontology at update time by checking the consistency of all axiomatic rules, removing rules with the lowest confidence value until inconsistencies are eliminated (Volker *et al.*, 2008).

Probabilistic KBs are becoming increasingly popular due to their power to express uncertainty in human knowledge. In practical terms, they can usefully rank the correctness of knowledge extraction with a confidence factor calculated by probabilistic methods. Chen and Wang (2014) use an MLN to represent uncertain facts and rules in the KB called ProbKB. An MLN is a set of weighted first-order logic expressions, where weights indicate the degree of truth of the formula. Constraining rules are learned in the form of Horn clauses from web text. Inference on entities, relations and rules is done using a relational massively parallel processing database. A number of techniques were used for curating ProbKB, including functional and semantic constraints, and rule ranking (taking only the top significant rules). Results show much faster runtime compared with a similar probabilistic, relational KB. MLNs are also used in StatSnowball (Zhu *et al.*, 2009), as described above.

Another very large probabilistic KB, Google's Knowledge Vault (Dong *et al.*, 2014) assigns a probability of correctness to each triple (subject, predicate, object). Supervised machine learning is used for fusing prior knowledge from existing KBs (Freebase, YAGO, DeepDive, TAC competition methods) with extraction from the Web. The probability of correctness for each triple is calculated based on agreement between different extractors and priors.

4.3 Evaluating knowledge base construction

This section compares techniques for KB population and discusses the important issue of evaluating the techniques and measuring the quality of the resulting KBs.

A common way to evaluate the KB population 'task' is to compare results with a gold standard. While this method enables comparison of various techniques on the same footing, it suffers from two major

drawbacks: the huge amount of human work required to prepare gold standard data sets and unavoidable bias (Biemann, 2005). Ontology learning is difficult to evaluate and human annotators tend to disagree with one another, which then requires many annotators to complete the same task in order to apply agreement measures (Dellschaft & Staab, 2006).

The evaluation of both steps in KB population, extracting entities and relations, and fitting them into the ontology graph structure, can be made using common metrics from information retrieval. As pointed out by Maynard *et al.* (2008), however, precision, recall and F1 are not really sufficient for evaluation, as these metrics are based on a binary classification of results (counts of *true* or *false*). In ontology classification, credit should be given for a partial match, for example Researcher to Lecturer. Given a key and a response, Maynard and Peters propose a balance distance metric (BDM) that combines various counts from the ontology hierarchy, such as the distance from the most specific common parent to the root, the length of the shortest path to the common parent and the average chain length of all chains in the ontology containing key, response, and both key and response. Augmented metrics use BDM as a weight to define a weighted measure of precision, recall and F1. Experiments show that the augmented metrics give better differentiation of results when comparing learning algorithms.

Next, we present the comparison of results from the TAC competitions held in 2010, 2011 and 2013[7], comparing successful approaches to KB population in two categories: entity linking and slot filling. The conclusions drawn from these results are useful for comparing different methods, therefore we devote more attention to these reviews.

The task of *entity linking* requires matching a textual 'mention' of a named entity with a corresponding entry in the KB, or determining that such an entry does not exist (Ji *et al.*, 2010). Micro-average accuracy was used to measure task performance. All competing systems performed best on person entities and worst on geo-political entities. Most of the top 10 algorithms used supervised learning-based lexical levels and name tagging features, but the top two methods used non-supervised learners. Using semantic features improved the task performance by a few percentage points on average. Many methods used Wikipedia structured information (links, boxes) and reported improved performance.

The goal of *slot filling* is to provide missing attributes in a KB from a set of documents by giving the best matching document for each missing attribute. A reference KB was constructed from a Wikipedia snapshot. Results from this task were categorised by human assessors as correct, inexact, redundant or wrong. Summarising the results for 2010 based on Ji *et al.* (2010), the first thing to notice is that one of the methods (cortex) used an additional two million extensively annotated documents and achieved more than twice better performance compared with the second best. A variety of techniques were used by the successful methods, including distant learning, bootstrapping, question answering and rules. Most of the systems used a single technique, though one successful system, CUNY, adopted a hybrid approach. About 24% of slots required inference from multiple sentences, therefore required combining existing information retrieval approaches with distant learning and reasoning. Some slot filling methods used an external KB, such as Freebase, DBpedia and YAGO, but this did not improve their results much, as these KBs covered only a portion of entities used in the competition.

In the TAC 2011 competition (Ji *et al.*, 2011), a number of new trends emerged. First, more statistical methods were used, for matching name variants (statistical classification) and extracting topic features with topic modelling using LDA. Second, new learning algorithms, such as Random Forest and ListNet were introduced for ranking candidates, along with existing SVM for Ranking (SVMRank) and maximum entropy (MaxEnt). According to Chen and Ji (2011), ListNet achieves the best performance compared with SVMRank, MaxEnt and other algorithms using the same features and resources. Results obtained by one of the 2011 systems indicate that entity context and profile-based features can significantly improve the performance of person and organisation entity extraction, but decrease the performance of geo-political entity extraction, because global, not local, features are dominant in these entities. Another successful method for entity linking, similar to the integrated approach mentioned in Section 2.3, explored the idea of extracting and disambiguating multiple entities from many documents at the same time, expanding the KB and iterating this process.

[7] Comparisons of results for TAC 2012 and 2014 have not been published.

Two other methods from the same TAC 2011 competition are worth mentioning. Monahan *et al.* (2011) present entity linking using cross-document co-reference in a number of stages: (1) clustering mentions with identical normalised strings, (2) grouping mentions in each cluster from step 1 using a distance calculated with logistic regression, which resolves polysemy of mentions (mentions with different meaning would be more distant), (3) finding synonym mentions if they can be linked to the same KB or Wikipedia entry or embedded in a common phrase, and (4) linking of each cluster from stage 3 to a KB entry by voting of mentions in the cluster for a KB entry. Of significance due to the limited amount of work in this area on non-English languages, this paper evaluates two methods of linking multilingual mentions: translate and link to a KB and link to a native KB without translation. Interestingly, though as expected, linking native entities gives better performance. Zhang *et al.* (2011) also combine a number of techniques for entity linking: acronym expansion, instance selection and topic modelling (with LDA). In the first step, acronyms are expanded to full names using SVM trained on matching entities. Next, for each KB entry, name variants are produced from Wikipedia pages. Finally, generated candidates are ranked by an algorithm using a learning to rank technique and SVM.

TAC 2013 (Surdeanu, 2013) required that provenance information was included for both entity linking and slot filling. This is a reasonable requirement, as in many domains decisions have to be justified by giving the reason for a particular result. For English slot filling, the top results were achieved by Roth *et al.* (2013), mentioned in Section 3.3. The distant supervision used in this and other methods in recent years is a characteristic trend in data mining supported by KBs. The second top method, called RPI-BLENDER (Yu *et al.*, 2013), used multiple sources for truth verification and a knowledge graph for the best path resolution from discovered elements to slot filler nodes. It is interesting to notice that the F1 score for these methods, although higher than in previous years, is still relatively low (around 40%), indicating the difficulty of the task of automatic identification of missing parameters in KBs.

5 Utilisation of knowledge mining systems

Utilisation of a KB is an important step supporting the process of knowledge discovery from data. In this section, we describe a number of classes of KB usage: entity linking, question answering, summarisation, and event extraction and prediction. Applications may make use of reasoning over a KB to improve performance on these tasks, though the extent of the improvement due to reasoning is often not separately evaluated.

5.1 Entity linking

Entity linking is a main step in both KB population and usage. In the KB population task, entity linking allows identification of related entries in the KB and expansion of the KB with new candidate entities. For KB usage, such as query answering or summarisation, related entities from the KB are used to build a response. In Section 4.3, we referred to entity linking in the context of KB evaluation. Those methods, however, were designed and optimised specifically for the TAC competition. In this section, we describe a few more techniques used for entity linking that may be suitable for more general practical applications.

Stoyanov *et al.* (2012) use context documents for both mentions and KB entities. Given a document with entities to be linked, a set of context documents is produced, extended by Wikipedia documents (for news) or e-mail addresses (for e-mails). The entities from this set of documents are then matched with a similar set of pre-selected entities from the KBs by a resolver. The entities are then fused and the process is repeated until a desired confidence (or number of iterations) is reached. Details of the methods used for matching entities with the KB are not provided.

Twitter is an important source of information in many applications, because it brings the most up-to-date information in a streaming fashion, but processing tweets is a challenge due to their brevity and the language used. Recognising and matching concepts in tweets with KB entries facilitates querying and summarisation. Liu *et al.* (2013) use context to match entities from tweets with KB entries (Wikipedia is treated as a KB). Tweet entities are extracted by a named entity recogniser, and a paraphrase (name variations) dictionary is used to resolve spelling errors and abbreviations. The idea is to resolve a collection of entities simultaneously with matching KB entities using context. Similarity between entities

and mentions is calculated using a feature vector comprising features such as edit distance between mention and entity, occurrence of the entity in the title of a Wikipedia entry, similarity between tweets (cosine similarity, same or different Twitter account, containing the same hashtag, etc.). The results were favourably compared with two baseline methods that use Wikipedia for linking and disambiguation.

Other research on matching tweets with Wikipedia (called 'wikification') is described by Huang *et al.* (2014). Graph-based semi-supervised learning algorithms are used for the task. To extract meaningful mentions from tweets, information from multiple tweets is used for collective inference for both mention identification and disambiguation. All related mentions and their corresponding Wikipedia concepts connected by weights form a relational graph. Weights are calculated as a linear combination of scores from different types of relations. The method is evaluated on a small human-annotated data set of 502 tweets. Two further examples of linking tweets to news are as follows. In Guo *et al.* (2013), hashtags from tweets are compared with entities in news using a graph-based latent variable model. The performance of the method, however, is only a few per cent better than plain information retrieval. A more complex method presented in Hua *et al.* (2015) is built combining text segmentation and entity disambiguation with prior knowledge from Probase. Obtained accuracy results show significant improvement over disambiguation methods based on similar instances.

5.2 Question answering

The *question answering* task involves complex matching of semantic structures (entities and relations) with that of a KB. Once the matching is established, sub-concepts and super-concepts can be used for query answering.

In the Kosmix system (Deshpande *et al.*, 2013), query understanding starts with information extraction from the query using the KB as a dictionary. After resolving potential homonyms, the task returns a set of concepts from the KB mentioned in the query as the understanding of that query. Deep Web search works in similar way, except that the query refers to the web information sources linked with the KB. Mentions from the query are matched with these links and the resulting sources are combined to form the answer returned to the user.

Wang *et al.* (2015) treat queries as short texts and describes a method for short text understanding using KBs. This problem is similar to query answering in that the query needs to be understood (parsed semantically and linked to related concepts) before answering. Unlike news articles or other longer documents, short texts such as tweets do not contain sufficient information to perform semantic parsing or topic modelling separately. The approach taken in this paper is to organise terms, concepts and all relevant signals into a graph, and use an iterative random walk approach to arrive at an understanding of a text. Probase concepts are clustered into 5000 clusters, each of which is referred to as a concept itself. Non-instance words in the text, such as verbs and adjectives, are linked to concepts by a probabilistic bridging of these words to concepts in a large set of web documents. These elements are used in a semantic network, in which edges express the strength of relationships between elements. This enables a probabilistic mapping of terms from the query to concepts in the graph to be made by performing multiple rounds of random walk. The results show an advantage of the method over a number of approaches, such as Bayesian analysis for conceptual similarity, LDA for co-occurrence relationships with Probase and random walk.

Yao and Van Durme (2014) implemented a query answering mechanism based on information extraction combined with a web-scale corpus. In addition, NLP techniques like ReVerb are used in order to deal with the informality of queries. The system splits a query into entities, topics and question words. The topic words are used to match with the nodes of Freebase. All triples (arg1, predicate, arg2) related to the topic are extracted as a subset or a subgraph of the KB. The extraction task is defined as maximising the probability $P(R|Q)$ of a relation R, given question Q as a word vector, using a Naïve Bayes assumption. Results show an improvement in performance of 34% in F-measure (48 against 31%) against another Freebase semantic parser of Berant *et al.* (2013).

QALD (Question Answering over Linked Data), an initiative to compare question answering methods over knowledge graphs, is an open research challenge run for the last few years. The task is to answer a question,

which is given in natural English or another language. We briefly discuss the top methods from 2013 and 2014, starting with QALD 2013. The squall2sparql system proposed by Ferré (2013) translates a query from a controlled natural language for English, called SQUALL, to an intermediate logic representation and then to SPARQL[8]. The produced query can then be used with a SPARQL endpoint of a KB, returning results. The quall2sparql system achieves the best performance due to the fact that the original queries in natural language are manually translated into SQUALL, thus removing many ambiguities.

CASIA (He *et al.*, 2013) performs a transformation from natural language queries to SPARQL queries in order to improve matching to the DBpedia KB. The system first generates a set of query triples in the form <subject, predicate, object> and converts them to ontology triples. These triples are then converted into SPARQL queries and used to query a KB. Returned answers are scored and combined for the same score. Although precision and recall on the gold standard were low (35 and 36% on the test data set), the research shaped a baseline for later work. Other top performing systems such as Scalewelis, RTV and Intui2 (Cimiano *et al.*, 2013) used syntactic patterns and statistical approaches, including hidden Markov models and tree parsing engines.

The QALD 2014 challenge included three tasks: multilingual question answering, biomedical question answering over interlinked data and hybrid question answering from both structured and raw text from DBpedia. A discussion of results is presented by Unger *et al.* (2014). All submissions for the first task were in the English language only. The third task did not have any submissions. The best performer in the first task was the Xser system with a precision of 72%. Phrases are labelled with categories corresponding to KB entries, that is entity, relation or category, using a trained perceptron. The query intention represented as dependency between phrases is modelled as a directed acyclic graph, where nodes are phrases and edges are relationships. This builds a KB-independent structure which is then mapped to a concrete KB. The Xser system can be further improved by removing the need for human-created training examples and avoiding downstream error propagation.

The second best system in the question answering task was gAnswer, proposed by Zou *et al.* (2014). Similar to Xser, it uses a semantic query graph to model the intention of the question. This graph is matched with subgraphs of the KB, assuming they are in RDF format. Disambiguation of matching is done at the query evaluation stage, which is easier and saves on processing time. Therefore, this system achieves good runtime performance with about 1 second per query.

An alternative approach that does not make use of a general KB of facts to directly answer questions is exemplified by the IBM Watson system (Ferrucci, 2012). The approach taken is to dynamically search a large number of data sources in order to find an answer to a given question. The aim of the system is to efficiently extract and evaluate candidate answers in order to find the most likely answer, using learned models to weigh competing evidence. Watson uses a 'knowledge base' called PRISMATIC, organised as a bag of frames, that contains domain-specific linguistic knowledge to assist in finding answers in these data sources (Fan *et al.*, 2012), for example that presidents win elections, countries annex regions, etc. This approach is feasible when answers to questions can be found without using extensive inference, but rather from direct matches between query expressions and text snippets in the data sources where the answers are contained.

There are a number of conclusions that can be drawn from this section. First, there is no simple technique that can address both entity matching and text understanding. Most successful methods use a holistic approach with iterated running of many stages of processing. Many successful solutions with different techniques achieve similar results. Second (and related to the first), including context always helps. The context comes in different forms: processing many entities at the same time, or using a large set of web documents or an external large KB. Therefore, the choice of methods for a particular problem depends on the availability of data sources, ease of implementation and other requirements such as runtime. Finally, a system should be able to be rapidly adapted to incorporate new data sources and new types of knowledge.

5.3 Summarisation

In this brief section we consider the use of KBs for automatic summarisation of text, for example news articles. We cannot locate any research methods that explicitly use KBs for article summarisation, but

[8] A semantic query language for retrieving and manipulating data in RDF format.

looking at many existing summarisation techniques, we believe that such use could be beneficial. This is because summarisation often relies on additional context (Aggarwal & Zhai, 2012: Ch. 3), such as links to other information or contents of user queries. A KB could provide such context. Other summarisation methods, such as sentence scoring (Nenkova & McKeown, 2012), clustering and topic models also have the potential to improve when using a KB. These hypotheses, however, need to be tested in future research.

A related problem is entity summarisation where information, including a timeline, is provided for a given entity, such as a person or organisation. Such summaries can be generated automatically or combining manual and automatic generation. The work of Althoff *et al.* (2015) is an example of the latter approach, focussing specifically on generating a timeline for a given entity of interest using a KB with timestamped facts, such as Freebase. In the first step, candidate events for the entity are extracted from the KB by exploring related entities. In the second step, events are selected from these candidates using entity and date relevance measures. Events with the same entity and timestamp are merged to form compound events. Event descriptions are created in part manually by defining 100 frequent templates, and for the rest by concatenating the English names of the corresponding predicates and entities.

5.4 Event extraction and prediction

In Section 3.4, we discussed the general issue of event detection and tracking. In this section, we describe two event detection methods that use KBs.

Rusu *et al.* (2014) implement a method for extracting events from news articles with the use of the BabelNet KB. They use an unsupervised information extraction technique to discover events, in contrast to supervised classification of events into pre-defined types. Verbs identified as event triggers and their arguments are extracted from a dependency tree produced by ZPar, the dependency parser[9]. WordNet super-senses, BabelNet senses and hypernyms are used for event argument disambiguation. Events are clustered using the Chinese Whispers algorithm and a distance measure based on word counts in the sentence and super-senses from WordNet to obtain a generalised representation of events. Evaluation is performed by manually assessing a number of extracted events for completeness.

Asr *et al.* (2014) apply an ontology for event co-reference. The task is to cluster events in various news articles that are referred to together. The ontology consists of three layers: event classes, for example killing, earthquake; event instances; and event mentions in news. Each class of events has its own set of specific attributes. The ultimate goal is to match mentions from news with the class and instance of event. Class identification is performed with the help of WordNet synsets. Instance identification within the class is done by matching event attributes against the news. It would be useful to compare the results of event clustering with and without the use of an ontology, however, results are not provided in this paper.

6 Conclusion and further research directions

In this survey, we provided a concise state-of-the-art review of potential interest to researchers and knowledge engineers constructing knowledge mining systems. Despite a multitude of publications on data mining and NLP, most of this research is fragmented, covering only a small niche of larger research problems and not considering integration with other methods: one of our main objectives has been to draw together this disparate work to identify possible synergies and future trends. In many applications, research seeks to provide integrated and efficient solutions to accurately identifying and describing events, entities and topics, mined from open data sources, to facilitate strategic planning and/or preventive actions. These sources are streaming in nature and require efficient solutions to deal with the speed of information flow and concept drift. Another of our objectives has been to highlight the suitability of current methods for handling high-velocity data streams.

Reviewing the state-of-the-art in the area of knowledge mining and KB construction reveals a number of general trends. The first trend is a greater reliance on semantic contexts and integrated approaches to knowledge extraction. We have seen, for example, that methods linking multiple mentions from whole

[9] http://sourceforge.net/projects/zpar/

sentences and documents, or even larger contexts, achieve better results than those performing tasks in isolation. Bag-of-words methods, although still used as a baseline, are rarely the best approach to more advanced processing.

A second general trend is the proliferation of probabilistic approaches that have gained increasing popularity in recent years, starting with Bayesian reasoning and LDA, through CRF and a variety of models based on Markov chains, to complex graph analysis. Using these methods requires careful selection, as they differ in terms of performance and runtime and may depend on data set characteristics.

Another problem that we clearly identified is that most of the methods described in the literature, despite the claims that they are fit for online applications, were tested in offline mode, on limited, human-annotated data sets. In our experience, these methods require much more testing in continuous streaming mode that may uncover potential problems not foreseen by their original designers. Moreover, these methods were tested using different metrics and compared with a variety of baseline techniques, many of which are very simple, such as *tf-idf* for feature extraction. This makes it difficult to compare them to one another to select the most suitable method for further evaluation. Another problem is to recognise the bias in each method and find out, mostly experimentally, which of them works best in a given domain.

Out of many research opportunities evident from this review, we highlight those related to integrating knowledge mining from multiple data streams with the use of KBs to uncover a richer set of features. Interleaving many tasks in an iterative process, such as entity extraction and resolution supported by a KB, may help strengthen the confidence of discovered associations between entities and relations based on their interdependence, especially in the understanding of unstructured short texts such as tweets.

The following points summarise other research directions and technological gaps identified.

- In most recent publications, we have not seen many attempts to extract entities and relations from streaming data sources. We note that some systems, such as Kosmix (Chai *et al.*, 2013) and Rosette (Clarke *et al.*, 2012) have such capabilities, but many details are proprietary.
- Continued use of a KB over an extended period of time raises the issue of KB dynamics, which has not been adequately addressed in work on KB maintenance. Existing relations between given entities in a KB may cease to exist, or instead other relations may arise. For example, a person may start working for a different organisation. Inference mechanisms should be powerful enough to ascertain that the person stopped working for a previous organisation.
- A related issue is temporal reasoning, the problem being how to add temporal information to the triples commonly used to represent knowledge, then use this efficiently in reasoning. Continuing the example of the person changing jobs, a KB might record that the person worked for the previous organisation until the date of starting the new job. KB queries could also include a temporal parameter to refer to such dates. Clearly, temporal information is needed for summarisation applications involving event timelines. Hoffart *et al.* (2013) address this problem in the YAGO2 KB by introducing a temporal annotation to entities, but more work is needed to formalise this approach and especially to link events with KB entities.
- More work is needed to address concept drift in contexts where more training examples are created automatically from an initial set of labelled cases or from existing KBs or other sources using distant learning or bootstrapping techniques. This requires update and maintenance of KBs in near real time. In a fast-flowing information environment, these tasks have to catch up with both the data stream and user demands for up-to-date information.
- Aggregation of human knowledge and feedback into KB construction seems inevitable despite the trend towards automation. Further research is needed to find effective methods for integrating user feedback into KB construction to improve system accuracy and tackle concept drift in data streams.
- Entity linking and question answering in non-English languages has been noted in the summary papers of TAC 2011 and QALD 2012 (Cimiano *et al.*, 2013; Unger *et al.*, 2014) as a challenge not often addressed by researchers. In particular, it has been noted that linking native entities gives better performance than translating them to English before linking (Monahan *et al.*, 2011). Therefore, the question remains if, and how much better, results can be achieved by building a KB from a native language rather than relying on translation and using a KB built from English. A serious problem is linking together or merging multiple KBs constructed from different languages.

- Results from international document processing and knowledge extraction competitions, such as TAC and QALD, are valuable resources for comparing and selecting methods for evaluation and extension. Further research and testing is needed, however, to ascertain their applicability and performance in real applications and in domains other than those they were originally tested on.
- Another open research area is ontology merging. The increasing importance of this issue is due to the fact that there are many open ontologies currently available. Karma (an information integration tool) supports merging ontologies, though currently only on a small scale.
- Combining specific interestingness measures with event detection, as indicated in Section 3.4, may help in discovering rare events or events of interest that are outside pre-defined, known classes. Similar to clustering in Li *et al.* (2011), this may help reduce search dimensionality.
- In Section 5, we identified the potential of using KBs in text summarisation by providing a context for topic models and sentence classification. There is also a much greater scope for reusing knowledge accumulated in a KB for event identification. For example, in the EMBERS system (Ramakrishnan *et al.*, 2014), event prediction relies on processing vast volumes of data. A KB could be built in the background process of stream mining and then used in (or instead of) dynamic query expansion and fusion tasks. One challenge would be to find what kind of knowledge would be useful to accumulate in a KB and reuse, given that the event horizon changes rapidly with time.

Addressing the above research problems will allow a more extensive use of KBs for summarisation and event prediction alongside traditional methods for knowledge mining, such as clustering and classification.

Acknowledgement

This work was supported by Data to Decisions Cooperative Research Centre.

References

Agarwal, A., Chapelle, O., Dudík, M. & Langford, J. 2014. A reliable effective terascale linear learning system. *Journal of Machine Learning Research* **15**, 1111–1133.

Aggarwal, C. C. & Zhai, C. 2012. *Mining Text Data*. Springer.

Agichtein, E. & Gravano, L. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, 85–94.

Agrawal, R. & Srikant, R. 1995. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, 3–14.

Althoff, T., Dong, X. L., Murphy, K., Alai, S., Dang, V. & Zhang, W. 2015. TimeMachine: timeline generation for knowledge-base entities. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 19–28.

Angeli, G., Gupta, S., Premkumar, M. J., Manning, C. D., Ré, C., Tibshirani, J., Wu, J. Y., Wu, S. & Zhang, C. 2014. Stanford's distantly supervised slot filling systems for KBP 2014. In *Proceedings of the Seventh Text Analysis Conference*.

Antoniou, G. & van Harmelen, F. 2009. Web ontology language (OWL). In *Handbook on Ontologies*, Staad S. & Studer R. (eds). Springer, 91–110.

Asr, F. T., Sonntag, J., Grishina, Y. & Stede, M. 2014. Conceptual and practical steps in event coreference analysis of large-scale data. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference and Representation*, 35–44.

Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R. & Morales-Bueno, R. 2004. Early drift detection method. In *Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams*, 77–86.

Becker, H., Iter, D., Naaman, M. & Gravano, L. 2012. Identifying content for planned events across social media sites. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, 533–542.

Becker, H., Naaman, M. & Gravano, L. 2011. *Beyond Trending Topics: Real-World Event Identification on Twitter*. Technical report CUCS-012-11, Department of Computer Science, Columbia University.

Beltagy, I., Erk, K. & Mooney, R. 2014. Probabilistic soft logic for semantic textual similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1210–1219.

Berant, J., Chou, A., Frostig, R. & Liang, P. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1533–1544.

Biemann, C. 2005. Ontology learning from text: a survey of methods. *Journal for Language Technology and Computational Linguistics* **20**, 75–93.

Bifet, A. & Gavaldà, R. 2006. Learning from time-changing data with adaptive windowing. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 443–448.

Blei, D. M., Ng, A. Y. & Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* **3**, 993–1022.

Bollacker, K., Tufts, P., Pierce, T. & Cook, R. 2007. A platform for scalable, collaborative, structured information integration. In *Proceedings of the Sixth International Workshop on Information Integration on the Web*, 22–27.

Bröcheler, M., Mihalkova, L. & Getoor, L. 2010. Probabilistic similarity logic. In *Proceedings of the Twenty-Sixth Annual Conference on Uncertainty in Artificial Intelligence*, 73–82.

Brunzel, M 2008. The XTREEM methods for ontology learning from web documents. In *Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, Buitelaar P. & Cimiano P. (eds). IOS Press, 3–26.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R. & Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1306–1313.

Chai, X., Deshpande, O., Garera, N., Gattani, A., Lam, W., Lamba, D. S., Liu, L., Tiwari, M., Tourn, M., Vacheri, Z., Prasad, S. T. S., Subramaniam, S., Harinarayan, V., Rajaraman, A., Ardalan, A., Das, S., Suganthan G. C., P. & Doan, A. 2013. Social media analytics: the Kosmix story. *IEEE Data Engineering Bulletin* **36**, 4–12.

Chen, Y. & Wang, D. Z. 2014. Knowledge expansion over probabilistic knowledge bases. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 649–660.

Chen, Z. & Ji, H. 2011. Collaborative ranking: a case study on entity linking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 771–781.

Cheng, Z., Caverlee, J. & Lee, K. 2010. You are where you tweet: a content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 759–768.

Cimiano, P., Lopez, V., Unger, C., Cabrio, E., Ngomo, A.-C. N. & Walter, S. 2013. Multilingual Question Answering over Linked Data (QALD-3): lab overview. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, Forner P., Müller H., Paredes R., Rosso P. & Stein B. (eds). Springer-Verlag, 321–332.

Clarke, J., Merhav, Y., Suleiman, G., Zheng, S. & Murgatroyd, D. 2012. Basis technology at TAC 2012 entity linking. In *Proceedings of the Fifth Text Analysis Conference*.

Compton, P. & Jansen, R. 1990. A philosophical basis for knowledge acquisition. *Knowledge Acquisition* **2**, 241–258.

Cortes, C. & Vapnik, V. 1995. Support-vector networks. *Machine Learning* **20**, 273–297.

Curran, J. R., Murphy, T. & Scholz, B. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the Tenth Conference of the Pacific Association for Computational Linguistics*, 172–180.

Davis, A., Veloso, A., da Silva, A. S., Meira,, W. J. & Laender, A. H. F. 2012. Named entity disambiguation in streaming data. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers* , **1**, 815–824.

Dellschaft, K. & Staab, S. 2006. On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of the 5th International Conference on the Semantic Web*, 228–241.

Deshpande, O., Lamba, D. S., Tourn, M., Das, S., Subramaniam, S., Rajaraman, A., Harinarayan, V. & Doan, A. 2013. Building, maintaining, and using knowledge bases: a report from the trenches. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 1209–1220.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S. & Zhang, W. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 601–610.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. & Yates, A. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence* **165**, 91–134.

Fan, J., Kalyanpur, A., Gondek, D. C. & Ferrucci, D. A. 2012. Automatic knowledge extraction from documents. *IBM Journal of Research and Development* **56**, 5:1–5:10.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. 1996. From data mining to knowledge discovery in databases. *AI Magazine* **17**, 37–54.

Ferré, S. 2013. Squall2sparql: a translator from controlled English to full SPARQL 1.1. In *Proceedings of the Question Answering over Linked Data (QALD-3)*.

Ferrucci, D. A. 2012. Introduction to 'This is Watson'. *IBM Journal of Research and Development* **56**, 1:1–1:15.

Fung, G. P. C., Yu, J. X., Yu, P. S. & Lu, H. 2005. Parameter free bursty events detection in text streams. In *Proceedings of the 31st International Conference on Very Large Data Bases*, 181–192.

Furht, B. & Escalante, A. 2011. *Handbook of Data Intensive Computing*. Springer Science & Business Media.

Gama, J. 2012. A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence* **1**, 45–55.

Gama, J., Medas, P., Castillo, G. & Rodrigues, P. 2004. Learning with drift detection. In *Advances in Artificial Intelligence*, Bazzan A. L. C. & Labidi S. (eds). Springer-Verlag, 66–112.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. & Bouchachia, A. 2014. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)* **46**, 44.

Gao, D., Li, X. C. W., Zhang, R. & Ouyang, Y. 2014. Sequential summarization: a full view of Twitter trending topics. *IEEE Transactions on Knowledge and Data Engineering* **22**, 296–302.

Gattani, A., Lamba, D. S., Garera, N., Tiwari, M., Chai, X., Das, S., Subramaniam, S., Rajaraman, A., Harinarayan, V. & Doan, A. 2013. Entity extraction, linking, classification, and tagging for social media: a Wikipedia-based approach. *Proceedings of the VLDB Endowment* **6**, 1126–1137.

Geng, L. & Hamilton, H. J. 2006. Interestingness measures for data mining: a survey. *ACM Computing Surveys (CSUR)* **38**, 1–32.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* **5**, 199–220.

Guo, W., Li, H., Ji, H. & Diab, M. T. 2013. Linking tweets to news: a framework to enrich short text data in social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 239–248.

Gupta, A., Mumick, I. S. & Subrahmanian, V. S. 1993. Maintaining views incrementally. *ACM SIGMOD Record* **22**, 157–166.

Han, J., Kamber, M. & Pei, J. 2011. *Data Mining: Concepts and Techniques*. Elsevier.

He, S., Liu, S., Chen, Y., Zhou, G., Liu, K. & Zhao, J. 2013. CASIA@QALD-3: a question answering system over linked data. In *Proceedings of the Question Answering over Linked Data (QALD-3).*

Ho, V. H., Wobcke, W. & Compton, P. 2003. EMMA: an e-mail management assistant. In *Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology*, 67–74.

Hoffart, J., Suchanek, F. M., Berberich, K. & Weikum, G. 2013. YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence* **194**, 28–61.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L. & Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 541–550.

Hua, W., Wang, Z., Wang, H., Zheng, K. & Zhou, X. 2015. Short text understanding through lexical-semantic analysis. In *2015 IEEE 31st International Conference on Data Engineering (ICDE)*, 495–506.

Huang, H., Cao, Y., Huang, X., Ji, H. & Lin, C.-Y. 2014. Collective tweet wikification based on semi-supervised graph regularization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 380–389.

Huang, R. & Riloff, E. 2013. Multi-faceted event recognition with bootstrapped dictionaries. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 41–51.

Hulten, G., Spencer, L. & Domingos, P. 2001. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 97–106.

Ji, H. & Grishman, R. 2011. Knowledge base population: successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, **1**, 1148–1158.

Ji, H., Grishman, R. & Dang, H. T. 2011. Overview of the TAC 2011 knowledge base population track. In *Proceedings of the Fourth Text Analysis Conference*.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K. & Ellis, J. 2010. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Third Text Analysis Conference*.

Kim, M. H. & Compton, P. 2012a. Improving open information extraction for informal web documents with ripple-down rules. In *Knowledge Management and Acquisition for Intelligent Systems*, Richards D. & Kang B. H. (eds). Springer-Verlag, 160–174.

Kim, M. H. & Compton, P. 2012b. Improving the performance of a named entity recognition system with knowledge acquisition. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management*, 97–113.

Kotov, A., Zhai, C. & Sproat, R. 2011. Mining named entities with temporally correlated bursts from multilingual web news streams. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 237–246.

Koychev, I. 2000. Gradual forgetting for adaptation to concept drift. In *Proceedings of the ECAI Workshop Current Issues in Spatio-Temporal Reasoning*, 101–106.

Krzywicki, A. & Wobcke, W. 2010. Exploiting concept clumping for efficient incremental e-mail categorization. In *Advanced Data Mining and Applications*, Cao L., Feng Y. & Zhong J. (eds). Springer-Verlag, 244–258.

Krzywicki, A. & Wobcke, W. 2011. Exploiting concept clumping for efficient incremental news article categorization. In *Advanced Data Mining and Applications*, Tang J., King I., Chen L. & Wang J. (eds). Springer-Verlag, 353–366.

Kumar, R., Raghavan, P., Rajagopalan, S. & Tomkins, A. 1999. Extracting large-scale knowledge bases from the web. In *Proceedings of the 25th International Conference on Very Large Data Bases*, 639–650.

Lafferty, J. D., McCallum, A. & Pereira, F. C. N. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289. Morgan Kaufmann Publishers.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* **10**, 707–710.

Li, J., Wang, G. A. & Chen, H. 2011. Identity matching using personal and social identity features. *Information Systems Frontiers* **13**, 101–113.

Li, Y., Wang, C., Han, F., Han, J., Roth, D. & Yan, X. 2013. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1070–1078.

Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F. & Lu, Y. 2013. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1304–1311.

Liu, X., Zhang, S., Wei, F. & Zhou, M. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, **1**, 359–367.

Maynard, D., Li, Y. & Peters, W. 2008. NLP techniques for term extraction and ontology population. In *Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, Buitelaar P. & Cimiano P. (eds). IOS Press, 107–127.

McGarry, K. 2005. A survey of interestingness measures for knowledge discovery. *The Knowledge Engineering Review* **20**, 39–61.

Mendes, P. N., Jakob, M. & Bizer, C. 2012. DBpedia: a multilingual cross-domain knowledge base. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, 1813–1817.

Mintz, M., Bills, S., Snow, R. & Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003–1011.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M. & Welling, J. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2302–2310.

Monahan, S. & Brunson, M. 2014. Qualities of eventiveness. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference and Representation*, 59–67.

Monahan, S., Lehmann, J., Nyberg, T., Plymale, J. & Jung, A. 2011. Cross-lingual cross-document coreference with entity linking. In *Proceedings of the Fourth Text Analysis Conference*.

Napoles, C., Gormley, M. & Van Durme, B. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, 95–100.

Nasukawa, T. & Nagano, T. 2001. Text analysis and knowledge mining system. *IBM Systems Journal* **40**, 967–984.

Nenkova, A. & McKeown, K. 2012. A Survey of Text Summarization Techniques. In *Mining Text Data*. Aggarwal C. C. and Zhai C. (eds). Springer Science+Business Media, 43–76.

Ottens, K., Aussenac-Gilles, N., Gleizes, M. P. & Camps, V. 2007. Dynamic ontology co-evolution from texts: principles and case study. In *Proceedings of the International Workshop on Emergent Semantics and Ontology Evolution*, 70–83.

Pan, J. Z. 2009. Resource description framework. In *Handbook on Ontologies*, Staad S. & Studer R. (eds). Springer, 71–90.

Park, S. S., Kim, Y. S. & Kang, B. H. 2004. Personalized web document classification using MCRDR. In *Proceedings of the Pacific Knowledge Acquisition Workshop 2004*, 63–73.

Pham, S. B. & Hoffmann, A. 2005. Incremental knowledge acquisition for extracting temporal relations. In *Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, 354–359.

Ramakrishnan, N., Butler, P., Muthiah, S., Self, N., Khandpur, R., Saraf, P., Wang, W., Cadena, J., Vullikanti, A., Korkmaz, G., Kuhlman, C., Marathe, A., Zhao, L., Hua, T., Chen, F., Lu, C.-T., Huang, B., Srinivasan, A., Trinh, K., Getoor, L., Katz, G., Doyle, A., Ackermann, C., Zavorin, I., Ford, J., Summers, K., Fayed, Y., Arredondo, J., Gupta, D. & Mares, D. 2014. 'Beating the news' with EMBERS: forecasting civil unrest using open source indicators. In *Proceedings of the Twentieth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1799–1808.

Ré, C., Sadeghian, A. A., Shan, Z., Shin, J., Wang, F., Wu, S. & Zhang, C. 2014. Feature Engineering for Knowledge Base Construction. *Data Engineering Bulletin* **37**, 26–40.

Riloff, E. & Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, 474–479.

Ritter, A., Clark, S., Mausam & Etzioni, O. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1524–1534.

Roth, B., Barth, T., Wiegand, M., Singh, M. & Klakow, D. 2013. Effective slot filling based on shallow distant supervision methods. In *Proceedings of the Sixth Text Analysis Conference*.

Rusu, D., Hodson, J. & Kimball, A. 2014. Unsupervised techniques for extracting and clustering complex events in news. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference and Representation*, 26–34.

Schrodt, P. A., Davis, S. G. & Weddle, J. L. 1994. Political science: KEDS—a program for the machine coding of event data. *Social Science Computer Review* **12**, 561–587.

Shin, J., Wu, S., Wang, F., Sa, C. D., Zhang, C. & Ré, C. 2015. Incremental knowledge base construction using DeepDive. *Proceedings of the VLDB Endowment* **8**, 1310–1321.

Silva, L. D. & Riloff, E. 2014. User type classification of tweets with implications for event recognition. In *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, 98–108.

Stoyanov, V., Xu, J., Oard, D., Lawrie, D. & Finin, T. 2012. A context-aware approach to entity linking. In *Proceedings of the NAACL Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction*, 62–67.

Suganthan, G. C, Sun, P. C., Krishna Gayatri, K., Zhang, H., Yang, F., Rampalli, N., Prasad, S., Arcaute, E., Krishnan, G., Deep, R., Raghavendra, V. & Doan, A. 2015. Why big data industrial systems need rules and what we can do about it. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 265–276.

Surdeanu, M. 2013. Overview of the TAC 2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Proceedings of the Sixth Text Analysis Conference*.

Tudorache, T., Noy, N. F., Tu, S. & Musen, M. A. 2008. Supporting collaborative ontology development in protégé. In *The Semantic Web – ISWC 2008*, Sheth A., Staab S., Dean M., Paolucci M., Maynard D., Finin T. & Thirunarayan K. (eds). Springer-Verlag, 17–32.

Unger, C., Forascu, C., Lopez, V., Ngomo, A.-C. N., Cabrio, E., Cimiano, P. & Walter, S. 2014. . *Question Answering over Linked Data (QALD-4)*. CLEF 2014 Working Notes, 1172–1180.

Van Dyke Parunak, H., Rohwer, R., Belding, T. & Brueckner, S. 2007. Dynamic decentralized any-time hierarchical clustering. In *Engineering Self-Organising Systems*, Brueckner S., Hassas S., Jelasity M. & Yamins D. (eds). Springer-Verlag, 66–81.

Veloso, A., Meira, W. Jr. & Zaki, M. J. 2006. Lazy associative classification. In *Proceedings of the Sixth International Conference on Data Mining*, 645–654.

Volker, J., Haase, P. & Hitzler, P. 2008. Learning expressive ontologies. In *Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, Buitelaar P. & Cimiano P. (eds). IOS Press, 45–69.

Wang, Z., Zhao, K., Wang, H., Meng, X. & Wen, J.-R. 2015. Query understanding through knowledge-based conceptualization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3264–3270.

Widmer, G. 1997. Tracking context changes through meta-learning. *Machine Learning* **27**, 259–286.

Witten, I. H., Frank, E. & Hall, M. A. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd edition. Morgan Kaufmann Publishers.

Wobcke, W., Krzywicki, A. & Chan, Y.-W. 2008. A large-scale evaluation of an e-mail management assistant. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 438–442.

Yang, Y., Carbonell, J. G., Brown, R. D., Pierce, T., Archibald, B. T. & Liu, X. 1999. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems* **14**, 32–43.

Yao, X. & Van Durme, B. 2014. Information extraction over structured data: question answering with Freebase. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 956–965.

Yu, D., Li, H., Cassidy, T., Li, Q., Huang, H., Chen, Z., Ji, H., Zhang, Y. & Roth, D. 2013. RPI-BLENDER TAC-KBP2013 knowledge base population system. In *Proceedings of the Sixth Text Analysis Conference*.

Zacks, J. M. & Tversky, B. 2001. Event structure in perception and conception. *Psychological Bulletin* **127**, 3–21.

Zhang, W., Su, J., Chen, B., Wang, W., Toh, Z., Sim, Y., Cao, Y., Lin, C. Y. & Tan, C. L. 2011. I2R-NUS-MSRA at TAC 2011: entity linking. In *Proceedings of the Fourth Text Analysis Conference*.

Zhu, J., Nie, Z., Liu, X., Zhang, B. & Wen, J.-R. 2009. StatSnowball: a statistical approach to extracting entity relationships. In *Proceedings of the 18th International Conference on World Wide Web*, 101–110.

Zou, L., Huang, R., Wang, H., Yu, J. X., He, W. & Zhao, D. 2014. Natural language question answering over RDF: a graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 313–324.