NLP Techniques for Term Extraction and Ontology Population

Diana MAYNARD [1], Yaoyong LI and Wim PETERS
Dept. of Computer Science, University of Sheffield, UK

Abstract.
This chapter investigates NLP techniques for ontology population, using a combination of rule-based approaches and machine learning. We describe a method for term recognition using linguistic and statistical techniques, making use of contextual information to bootstrap learning. We then investigate how term recognition techniques can be useful for the wider task of information extraction, making use of similarity metrics and contextual information. We describe two tools we have developed which make use of contextual information to help the development of rules for named entity recognition. Finally, we evaluate our ontology-based information extraction results using a novel technique we have developed which makes use of similarity-based metrics first developed for term recognition.

Keywords. information extraction, ontology population, term recognition,

1. Introduction

In semantic web applications, ontology development and population are tasks of paramount importance. The manual performance of these tasks is labour- and therefore cost-intensive, and would profit from a maximum level of automation. For this purpose, the identification and extraction of terms that play an important role in the domain under consideration, is a vital first step.

Automatic term recognition (also known as term extraction) is a crucial component of many knowledge-based applications such as automatic indexing, knowledge discovery, terminology mining and monitoring, knowledge management and so on. It is particularly important in the healthcare and biomedical domains, where new terms are emerging constantly.

Term recognition has been performed on the basis of various criteria. The main distinction we can make is between algorithms that only take the distributional properties of terms into account, such as frequency and tf/idf [1], and extraction techniques that use the contextual information associated with terms. The work described here concentrates on the latter task, and describes algorithms that compare and measure context vectors, exploiting semantic similarity between terms and candidate terms. We then proceed to investigate a more general method for information extraction, which is used, along with term extraction, for the task of ontology population.

[1]Corresponding Author: Diana Maynard: Dept. of Computer Science, University of Sheffield, 211 Portobello St, Sheffield, UK; E-mail: diana@dcs.shef.ac.uk

Ontology population is a crucial part of knowledge base construction and maintenance that enables us to relate text to ontologies, providing on the one hand a customised ontology related to the data and domain with which we are concerned, and on the other hand a richer ontology which can be used for a variety of semantic web-related tasks such as knowledge management, information retrieval, question answering, semantic desktop applications, and so on.

Ontology population is generally performed by means of some kind of ontology-based information extraction (OBIE). This consists of identifying the key terms in the text (such as named entities and technical terms) and then relating them to concepts in the ontology. Typically, the core information extraction is carried out by linguistic pre-processing (tokenisation, POS tagging etc.), followed by a named entity recognition component, such as a gazetteer and rule-based grammar or machine learning techniques. Named entity recognition (using such approaches) and automatic term recognition are thus generally performed in a mutually exclusive way: i.e. one or other technique is used depending on the ultimate goal. However, it makes sense to use a combination of the two techniques in order to maximise the benefits of both. For example, term extraction generally makes use of frequency-based information whereas typically named entity recognition uses a more linguistic basis. Note also that a "term" refers to a specific concept characteristic of a domain, so while a named entity such as Person or Location is generic across all domains, a technical term such as "myocardial infarction" is only considered a relevant term when it occurs in a medical domain: if we were interested in sporting terms then it would probably not be considered a relevant term, even if it occurred in a sports article. As with named entities, however, terms are generally formed from noun phrases (in some contexts, verbs may also be considered terms, but we shall ignore this here).

The overall structure of the chapter covers a step by step description of the natural task extension from term extraction into more general purpose information extraction, and therefore brings together the whole methodological path from extraction, through annotation to ontology population.

2. A Similarity-based Approach to Term Recognition

The TRUCKS system [2] introduced a novel method of term recognition which identified salient parts of the context surrounding a term from a variety of sources, and measured their strength of association with relevant candidate terms. This was used in order to improve on existing methods of term recognition such as the C/NC-Value approach [3] which used largely statistical methods, plus linguistic (part-of-speech) information about the candidate term itself. The NC-Value method extended on the C-Value method by adding information about frequency of co-occurrence with context words. The SNC-Value used in TRUCKS includes contextual and terminological information and achieves improved precision (see [4] for more details).

In very small and/or specialised domains, as are typically used as a testbed for term recognition, statistical information may be skewed due to data sparsity. On the other hand, it is also difficult to extract suitable semantic information from such specialised corpora, particularly as appropriate linguistic resources may be lacking. Although contextual information has previously been used, e.g. in general language [5], and in the NC-Value method, only shallow semantic information is used in these cases. The TRUCKS

approach, however, identifies different elements of the context which are combined to form the Information Weight [2], a measure of how strongly related the context is to the candidate term. This Information Weight is then combined with statistical information about a candidate term and its context, acquired using the NC-Value method. Note that both approaches, unlike most other term recognition approaches, result in a ranked list of terms rather than making a binary decision about termhood. This introduces more flexibility into the application, as the user can decide at what level to draw the cut-off point. Typically, we found that the top 1/3 of the list produces the best results.

The idea behind using the contextual information stems from the fact that, just as a person's social life can provide valuable insight about their personality, so we can gather much information about a term by analysing the company it keeps. In general, the more similar context words are to a candidate term, the stronger the likelihood of the term being relevant. We can also use this same kind of criteria to perform term disambiguation, by choosing the meaning of the term closest to that of its context [6].

2.1. Acquiring Contextual Information

The TRUCKS system builds on the NC-Value method for term recognition, by incorporating contextual information in the form of additional weights. We acquire three different types of knowledge about the context of a candidate term: syntactic, terminological, and semantic. The NC Value method is first applied to the corpus to acquire an initial set of candidate terms.

Syntactic knowledge is based on *boundary words*, i.e. the words immediately before and after a candidate term. A similar method (the *barrier word* approach [7,8]) has been used previously to simply accept or decline the presence of a term, depending on the syntactic category of the barrier or boundary word. Our system takes this a stage further by - rather than making a binary decision - allocating a weight to each syntactic category based on a co-occurrence frequency analysis, to determine how likely the candidate term is to be valid. For example, a verb occurring immediately before a candidate term is statistically a much better indicator of a true term than an adjective is. By a "better indicator", we mean that a candidate term occurring with this context is more likely to be valid. Each candidate term is then assigned a syntactic weight, calculated by summing the category weights for all the context boundary words occurring with it.

Terminological knowledge concerns the terminological status of context words. A context word which is also a term (which we call a *context term*) is likely to be a better indicator of a term than one which is not also a term itself. This is based on the premise that terms tend to occur together. Context terms are determined by applying the NC-Value method to the whole corpus and selecting the top 30% of the resulting ranked list of terms. A context term (CT) weight is produced for each candidate term, based on its total frequency of occurrence with other context terms.

The CT weight is formally described as follows:

$$CT(a) = \sum_{d \epsilon T_a} f_a(d) \tag{1}$$

where
a is the candidate term,
T_a is the set of context terms of a,

d is a word from T_a,
$f_a(d)$ is the frequency of d as a context term of a.

Semantic knowledge is based on the idea of incorporating semantic information about terms in the context. We predict that context words which are not only terms, but also have a high degree of similarity to the candidate term in question, are more likely to be relevant. This is linked to the way in which sentences are constructed. Semantics indicates that words in the surrounding context tend to be related, so the more similar a word in the context is to a term, the more informative it should be.

Our claim is essentially that if a context word has some contribution towards the identification of a term, then there should be some significant correspondence between the meaning of that context word and the meaning of the term. This should be realised as some identifiable semantic relation between the two. Such a relation can be exploited to contribute towards the correct identification and comprehension of a candidate term. A similarity weight is added to the weights for the candidate term, which is calculated for each term / context term pair. This similarity weight is calculated using a new metric to define how similar a term and context term are, by means of their distance in a hierarchy. For the experiments carried out in [4], the UMLS semantic network was used [9].

While there exist many metrics and approaches for calculating similarity, the choice of measure may depend considerably on the type of information available and the intended use of the algorithm. A full discussion of such metrics and their suitability can be found in [4], so we shall not go into detail here. Suffice it to say that:

- Thesaurus-based methods seem a natural choice here, because to some extent they already define relations between words.
- Simple thesaurus-based methods fail to take into account the non-uniformity of hierarchical structures, as noted by [10].
- Methods such as information content [10] have the drawback that the assessment of similarity in hierarchies only involves taxonomic (is-a) links. This means that they may exclude some potentially useful information.
- General language thesauri such as WordNet and Roget's Thesaurus are only really suitable for general-language domains, and even then have been found to contain serious omissions. If an algorithm is dependent on resources such as this, it can only be as good as is dictated by the resource.

2.2. Similarity Measurement in the TRUCKS System

Our approach to similarity measurement in a hierarchy is modelled mainly on the EBMT (Example-Based Machine Translation)-based techniques of Zhao [11] and Sumita and Iida [12]. This is based on the premise that the position of the MSCA (Most Specific Common Abstraction)[2] within the hierarchy is important for similarity. The lower down in the hierarchy the MSCA, the more specific it is, and therefore the more information is shared by the two concepts, thus making them more similar. We combine this idea with that of semantic distance [13,14,15]. In its simplest form, similarity is measured by edge-counting – the shorter the distance between the words, the greater their similarity. The MSCA is commonly used to measure this. It is determined by tracing the respective paths of the two words back up the hierarchy until a common ancestor is found. The

[2] also known as Least Common Subsumer or LCS

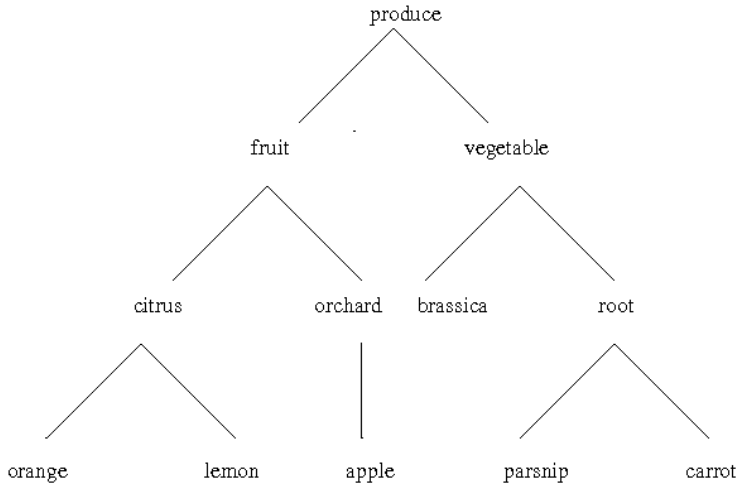

Figure 1. Fragment of a food network

average distance from node to MSCA is then measured: the shorter the distance to the MSCA, the more similar the two words. We combine these two ideas in our measure by calculating two weights: one which measures the distance from node to MSCA, and one which measures the vertical position of the MSCA. Note that this metric does of course have the potential drawback mentioned above, that only involving taxonomic links does mean the potential loss of information. However, we claim that this is quite minimal, due to the nature of the quite restricted domain-specific text that we deal with, because other kinds of links are not so relevant here. Futhermore, distance-based measures such as these are dependent on a balanced distribution of concepts in the hierarchy, so it is important to use a suitable ontology or hierarchy.

To explain the relationship between network position and similarity, we use the example of a partial network of fruit and vegetables, illustrated in Figure 1. Note that this diagram depicts only a simplistic is-a relationship between terms, and does not take into account other kinds of relationships or multidimensionality (resulting in terms occurring in more than one part of the hierarchy due to the way in which they are classified). We claim that the height of the MSCA is significant. The lower in the hierarchy the two items are, the greater their similarity. In the example, there would be higher similarity between *lemon* and *orange* than between *fruit* and *vegetable*. Although the average distance from *lemon* and *orange* to its MSCA (*citrus*) is the same as that from *fruit* and *vegetable* to its MSCA (*produce*), the former group is lower in the hierarchy than the latter group. This is also intuitive, because not only do *lemon* and *orange* have the *produce* feature in common, as *fruit* and *vegetable* do, but they also share the features *fruit* and *citrus*.

Our second claim is that the greater the horizontal distance between words in the network, the lower the similarity. By horizontal distance, we mean the distance between two nodes via the MSCA. This is related to the average distance from the MSCA, since the greater the horizontal distance, the further away the MSCA must be in order to be common to both. In the food example, *carrot* and *orange* have a greater horizontal distance than *lemon* and *orange*, because their MSCA (*produce*) is further away from them

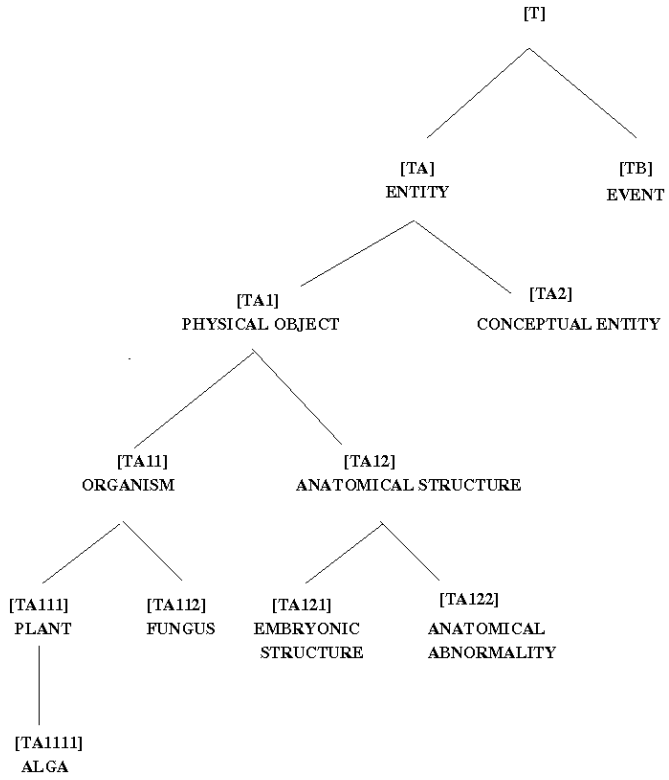

Figure 2. Fragment of the Semantic Network

than the MSCA of *lemon* and *orange* (*citrus*). Again, it is intuitive that the former are less similar than the latter, because they have less in common.

Taking these criteria into account, we define the following two weights to measure the vertical position of the MSCA and the horizontal distance between the nodes:

- *positional*: measured by the combined distance from root to each node
- *commonality*: measured by the number of shared common ancestors multiplied by the number of words (usually two).

The nodes in the Semantic Network are coded such that the number of digits in the code represents the number of leaves descended from the root to that node, as shown in Figure 2, which depicts a small section of the UMLS Semantic Network. Similarity between two nodes is calculated by dividing the commonality weight by the positional weight to produce a figure between 0 and 1, 1 being the case where the two nodes are identical, and 0 being the case where there is no common ancestor (which would only occur if there were no unique root node in the hierarchy). This can formally be defined as follows:

$$\mathrm{sim}(w_1 \ldots w_n) = \frac{\mathrm{com}(w_1 \ldots w_n)}{\mathrm{pos}(w_1 \ldots w_n)} \tag{2}$$

where

com($w_1 ... w_n$) is the commonality weight of words 1...n

pos($w_1 ... w_n$) is the positional weight of words 1...n.

It should be noted that the definition permits any number of nodes to be compared, although usually only two nodes would be compared at once. Also, it should be made clear that similarity is not being measured between terms themselves, but between the semantic types (concepts) to which the terms belong. So a similarity of 1 indicates not that two terms are synonymous, but that they both belong to the same semantic type.

3. Moving from Term to Information Extraction

There is a fairly obvious relationship between term recognition and information extraction, the main difference being that information extraction may also look for other kinds of information than just terms, and it may not necessarily be focused on a specific domain. Traditionally, methods for term recognition have been strongly statistical, while methods for information extraction have focused largely on either linguistic methods or machine learning, or a combination of the two. Linguistic methods for information extraction (IE), such as those used in GATE [16], are generally rule-based, and in fact use methods quite similar to those for term extraction used in the TRUCKS system, in that they use a combination of gazetteer lists and hand-coded pattern-matching rules which use contextual information to help determine whether such "candidate terms" are valid, or to extend the set of candidate terms. We can draw a parallel between the use of gazetteer lists containing sets of "seed words" and the use of candidate terms in TRUCKS: the gazetteer lists act as a starting point from which to establish, reject, or refine the final entity to be extracted.

3.1. Information Extraction with ANNIE

GATE, the General Architecture for Text Engineering, is a framework providing support for a variety of language engineering tasks. It includes a vanilla information extraction system, ANNIE, and a large number of plugins for various tasks and applications, such as ontology support, information retrieval, support for different languages, WordNet, machine learning algorithms, and so on. There are many publications about GATE and ANNIE – see for example [17]. This is not the focus of this paper, however, so we simply summarise here the components and method used for rule-based information extraction in GATE.

ANNIE consists of the following set of processing resources: tokeniser, sentence splitter, POS tagger, gazetteer, finite state transduction grammar and orthomatcher. The resources communicate via GATE's annotation API, which is a directed graph of arcs bearing arbitrary feature/value data, and nodes rooting this data into document content (in this case text).

The **tokeniser** splits text into simple tokens, such as numbers, punctuation, symbols, and words of different types (e.g. with an initial capital, all upper case, etc.), adding a "Token" annotation to each. It does not need to be modified for different applications or text types.

The **sentence splitter** is a cascade of finite-state transducers which segments the text into sentences. This module is required for the tagger. Both the splitter and tagger are generally domain and application-independent.

The **tagger** is a modified version of the Brill tagger, which adds a part-of-speech tag as a feature to each Token annotation. Neither the splitter nor the tagger is a mandatory part of the NE system, but the annotations they produce can be used by the semantic tagger (described below), in order to increase its power and coverage.

The **gazetteer** consists of lists such as cities, organisations, days of the week, etc. It contains some entities, but also names of useful key words, such as company designators (e.g. "Ltd."), titles (e.g. "Dr."), etc. The lists are compiled into finite state machines, which can match text tokens.

The **semantic tagger** (or JAPE transducer) consists of hand-crafted rules written in the JAPE pattern language [18], which describe patterns to be matched and annotations to be created. Patterns can be specified by describing a specific text string or annotation (e.g. those created by the tokeniser, gazetteer, document format analysis, etc.).

The **orthomatcher** performs coreference, or entity tracking, by recognising relations between entities. It also has a secondary role in improving NE recognition by assigning annotations to previously unclassified names, based on relations with existing entities.

ANNIE has been adapted to many different uses and applications: see [19,20,21] for some examples. In terms of adapting to new tasks, the processing resources in ANNIE fall into two main categories: those that are domain-independent, and those that are not. For example, in most cases, the tokeniser, sentence splitter, POS tagger and orthographic coreference modules fall into the former category, while resources such as gazetteers and JAPE grammars will need to be modified according to the application. Similarly, some resources, such as the tokeniser and sentence splitter, are largely language-independent (exceptions may include some Asian languages, for example), and some resources are more language-dependent, such as gazetteers.

3.2. Using contextual information to bootstrap rule creation

One of the main problems with using a rule-based approach to information extraction is that rules can be slow and time-consuming to develop, and an experienced language engineer is generally needed to create them. This language engineer typically needs also to have a detailed knowledge of the language and domain in question. Secondly, it is easy with a good gazetteer list and a simple set of rules to achieve reasonably accurate results in most cases in a very short time, especially where recall is concerned. For example, our work on surprise languages [20] achieved a reasonable level of accuracy on the Cebuano language with a week's effort and with no native speaker and no resources provided. Similarly, [22] achieved high scores for recognition of locations using only gazetteer lists. However, achieving very high precision requires a great deal more effort, especially for languages which are more ambiguous than English.

It is here that making use of contextual information is key to success. Gazetteer lists can go a long way towards initial recognition of common terms; a set of rules can boost this process by e.g. combining elements of gazetteer lists together, using POS information combined with elements of gazetteer lists (e.g. to match first names from a list with probable surnames indicated by a proper noun), and so on. In order to resolve

ambiguities and to find more complex entity types, context is necessary. Here we build on the work described in Section 2, which made use of information about contextual terms to help decide whether a candidate term (extracted initially through syntactic tagging) should be validated.

There are two tools provided in GATE which enable us to make use of contextual information: the gazetteer lists collector and ANNIC. These are described in the following two sections.

3.3. Gazetteer lists collector

The GATE gazetteer lists collector [23] helps the developer to build new gazetteer lists from an initial set of annotated texts with minimal effort. If the list collector is combined with a semantic tagger, it can be used to generate context words automatically. Suppose we generate a list of Persons occurring in our training corpus. Some of these Persons will be ambiguous, either with other entity types or even with non-entities, especially in languages such as Chinese. One way to improve Precision without sacrificing Recall is to use the lists collector to identify from the training corpus a list of e.g. verbs which typically precede or follow Persons. The list can also be generated in such a way that only verbs with a frequency above a certain threshold will be collected, e.g. verbs which occur less than 3 times with a Person could be discarded.

The lists collector can also be used to improve recognition of entities by enabling us to add constraints about contextual information that precedes or follows candidate entities. This enables us to recognise new entities in the texts, and forms part of a development cycle, in that we can then add such entries to the gazetteer lists, and so on. In this way, noisy training data can be rapidly created from a small seed corpus, without requiring a large amount of annotated data initially.

Furthermore, using simple grammar rules, we can collect not only examples of entities from the training corpus, but also information such as the syntactic categories of the preceding and following context words. Analysis of such categories can help us to write better patterns for recognising entities. For example, using the lists collector we might find that definite and indefinite articles are very unlikely to precede Person entities, so we can use this information to write a rule stipulating that if an article is found preceding a candidate Person, that candidate is unlikely to be a valid Person. We can also use lexical information, by collecting examples of verbs which typically follow a Person entity. If such a verb is found following a candidate Person, this increases the likelihood that such a candidate is valid, and we can assign a higher priority to such a candidate than one which does not have such context.

3.4. ANNIC

The second tool, ANNIC (ANNotations In Context) [24], enables advanced search and visualisation of linguistic information. This provides an alternative method of searching the textual data in the corpus, by identifying patterns in the corpus that are defined both in terms of the textual information (i.e. the actual content) and of metadata (i.e. linguistic annotation and XML/TEI markup). Essentially, ANNIC is similar to a KWIC (KeyWords In Context) index, but where a KWIC index provides simply text in context in response to a search for specific words, ANNIC additionally provides linguistic information (or other annotations) in context, in response to a search for particular linguistic patterns.

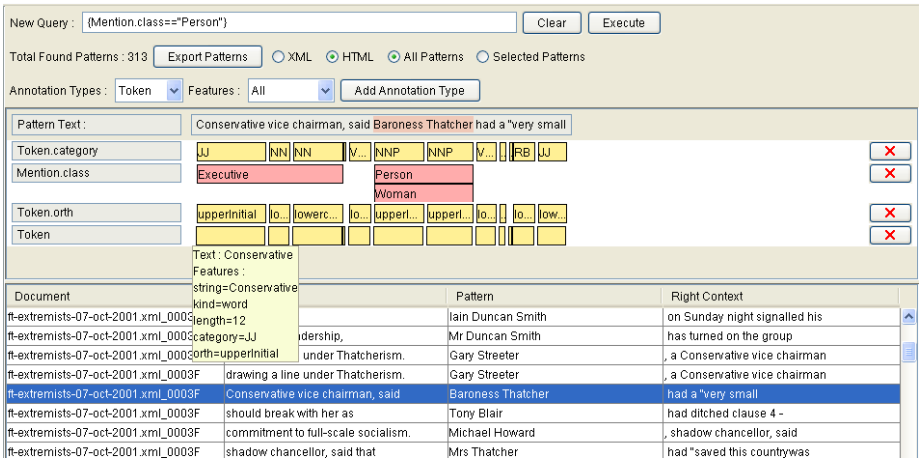

Figure 3. ANNIC Viewer

ANNIC can be used as a tool to help users with the development of JAPE rules by enabling them to search the text for examples using an annotation or combination of annotations as the keyword. Language engineers have to use their intuition when writing JAPE rules, trying to strike the ideal balance between specificity and coverage. This requires them to make a series of informed guesses which are then validated by testing the resulting ruleset over a corpus. ANNIC can replace the guesswork in this process with a live analysis of the corpus. Each pattern intended as part of a JAPE rule can easily be tested directly on the corpus and have its specificity and coverage assessed almost instantaneously.

Figure 3 shows a screenshot of ANNIC in use. The bottom section in the window contains the patterns along with their left and right context concordances, while the top section shows a graphical visualisation of the annotations. ANNIC shows each pattern in a separate row and provides a tool tip that shows the query that the selected pattern refers to. Along with its left and right context, it also lists the name of documents that the patterns come from. The tool is interactive, and different aspects of the search results can be viewed by clicking on appropriate parts of the GUI.

ANNIC can also be used as a more general tool for corpus analysis, because it enables querying the information contained in a corpus in more flexible ways than simple full-text search. Consider a corpus containing news stories that have been processed with a standard NE system such as ANNIE. A query like

{Organization} ({Token})*3 ({Token.string==’up’}|{Token.string==’down’}) ({Money} | {Percent})

would return mentions of share movements like “BT shares ended up 36p” or “Marconi was down 15%”. Locating this type of useful text snippets would be very difficult and time consuming if the only tool available were text search. Clearly it is not just information extraction and rule writing that benefits from the visualisation of contextual information in this way. When combined with the TRUCKS term extraction technique, we can use it to visualise the combinations of term and context term, and also to investigate other possible sources of interesting context which might provide insight into further refinement of the weights. We can also very usefully combine ANNIC with the

gazetteer list collector described in Section 3.3 in order to again visualise other sources of contextual information worth collecting.

4. From Traditional to Ontology-Based Information Extraction

Ontology-Based IE (OBIE) is one of the technologies used for semantic annotation, which is essentially about assigning to entities in the text links to their semantic descriptions. This sort of metadata provides both class and instance information about the entities. One of the important differences between traditional IE and OBIE is the use of a formal ontology rather than a flat lexicon or gazetteer structure. This may also involve reasoning.

4.1. OBIE Systems

There are a number of what we describe as ontology-oriented IE systems, which, unlike ontology-based ones, do not incorporate ontologies into the system, but either use them as a bridge between the IE output and the final annotation (as with AeroDAML) or rely on the user to provide the relevant information through manual annotation (as with the Amilcare-based tools).

AeroDAML [25] applies IE techniques to automatically generate DAML annotations from web pages. It links proper nouns and common types of relations with classes and properties in a DAML ontology. It makes use of an ontology in order to translate the extraction results into a corresponding RDF model.

Amilcare [26] is an IE system which has been integrated in several different semantic annotation tools, such as OntoMat [27], which combines a manual annotation tool with an IE system running in the background. It uses supervised rule learning to adapt to new domains and applications given human annotated texts (training data). It treats the semantic annotations as a flat set of labels, thus ignoring the further knowledge in the ontology. Amilcare uses GATE's NLP components in order to obtain linguistic information as features for the learning process.

One of the problems with these annotation tools is that they do not provide the user with a way to customise the integrated language technology directly. While many users would not need or want such customisation facilities, users who already have ontologies with rich instance data will benefit if they can make this data available to the IE components. However, this is not possible when traditional IE methods like Amilcare are used, because they are not aware of the existence of the user's ontology.

The more serious problem however, as discussed in the S-CREAM system [27], is that there is often a gap between the IE output annotations and the classes and properties in the user's ontology. The solution proposed by the developers was to write logical rules to resolve this. For example, an IE system would typically annotate `London` and `UK` as locations, but extra rules are needed to specify that there is a containment relationship between the two. However, rule writing of this kind is too difficult for most users and therefore ontology-based IE is needed, as it annotates directly with the classes and instances from the user's ontology.

In response to these problems, a number of OBIE systems have been developed. Magpie [28] is a suite of tools which supports semantic annotation of web pages. It is

fully automatic and works by matching the text against instances in the ontology. The SemTag system [29] is similar in approach to Magpie as it annotates texts by performing lookup against the TAP ontology. It also has a second, disambiguation phase, where SemTag uses a vector-space model to assign the correct ontological class or determine that this mention does not correspond to a class in TAP. The problem with both systems is that they are not able to discover new instances and are thus restricted in terms of recall.

The PANKOW system [30] exploits surface patterns and the redundancy on the Web to categorise automatically named entities found in text with respect to a given ontology. Its aim is thus primarily ontology population rather than annotation. PANKOW has recently been integrated with MAGPIE [31].

OntoSyphon [32] is similar to PANKOW and uses the ontology as the starting point in order to carry out web mining to populate the ontology with instances. It uses the ontology structure to determine the relevance of the candidate instances. However, it does not carry out semantic annotation of documents as such.

The KIM system [33] produces annotations linked both to the ontological class and to the exact individual in the instance base. For new (previously unknown) entities, new identifiers are allocated and assigned; then minimal descriptions are added to the semantic repository. KIM has a rule-based, human-engineered IE system based on GATE's ANNIE, which uses the ontology structure during pattern matching and instance disambiguation. The only shortcoming of this approach is that it requires human intervention in order to adapt it to new ontologies.

To summarise, all these systems use the ontology as their target output, and the ontology-based ones also use class and instance information during the IE process. While KIM and OntoSyphon do make use of the ontology structure, the former is a rule-based, not a learning approach, whereas the latter does not perform semantic annotation, only ontology population.

5. Evaluation of Ontology-Based Information Extraction

Traditionally, information extraction is evaluated using Precision, Recall and F-Measure. However, when dealing with ontologies, such methods are not really sufficient because they give us a binary decision of correctness, i.e. they classify the result as either right or wrong. This is fine for traditional IE, because an element identified as a Person is either correct or incorrect (measured by Precision), and elements which should be identified as Person are either identified or not (measured by Recall). When making an ontological classification, however, the distinction is a bit more fuzzy. For example if we misclassify an instance of a Researcher as a Lecturer, we are clearly less wrong than missing the identification (and classification) altogether, and we are also somehow less wrong than if we had misclassified the instance as a Location. Credit should therefore be given for partial correctness. Traditionally, this is sometimes achieved by allocating a half weight to something deemed partially correct, but this is still insufficient to give a proper distinction between degrees of correctness. We therefore adopt an approach based on similarity between Key (the gold standard) and Response (the output of the system).

5.1. A Distance-based Metric for Evaluation

We developed the Balanced Distance Metric (BDM) [34] in order to address this problem. This metric has been designed to replace the traditional "exact match or fail" metrics with a method which yields a graded correctness score by taking into account the semantic distance in the ontological hierarchy between the compared nodes (Key and Response).

The semantic distance is adapted from the semantic weight used in the TRUCKS system, but takes into account also some normalisation – something which was not considered in the original TRUCKS weight. In the BDM, each of the paths has been normalised with two additional measurements, of which the first is the average length of the chains in which key and response concepts occur. The longer a particular ontological chain is, the more difficult it is to consistently pick out a particular class for annotation [35]. The second normalization is the introduction of the branching factor (i.e. number of descendants) of the relevant nodes in the ontology. This is also an indication of the level of difficulty associated with the selection of a particular ontlogical class relative to the size of the set of candidates. These normalizations will make the penalty that is computed in terms of node traversal within our metric relative to the semantic density of the chains.

Another similar metric which has been proposed for this task is Learning Accuracy (LA) [36], which was originally developed to measure how well an item had been classified in an ontology. Learning Accuracy has a major flaw for our purposes, however, in that it does not take into account the depth of the key concept in the hierarchy, considering essentially only the height of the MSCA (Most Specific Common Abstraction) and the distance from the response to the MSCA. This means that however far away the key is from the MSCA, the metric will give the same outcome. The BDM is more balanced in this respect, because it takes the relative specificity of the taxonomic positions of the key and response into account in the score, but it does not distinguish between the specificity of the key concept on the one hand, and the specificity of the response concept on the other. For instance, the key can be a specific concept (e.g. 'car'), whereas the response can be a more general concept (e.g. 'relation'), or vice versa, and the result will be the same. This is not the case with the Learning Accuracy metric.

The BDM is computed on the basis of the following measurements:

- CP = the shortest length from root to the most specific common parent, i.e. the most specific ontological node subsuming both Key and Response)
- DPK = shortest length from the most specific common parent to the Key concept
- DPR = shortest length from the most specific common parent to the Response concept
- n1: average chain length of all ontological chains containing Key and Response.
- n2: average chain length of all ontological chains containing Key.
- n3: average chain length of all ontological chains containing Response.
- BR: the branching factor of each relevant concept, divided by the average branching factor of all the nodes from the ontology, excluding leaf nodes.

The complete BDM formula is as follows:

$$BDM = \frac{BR(CP/n1)}{BR(CP/n1) + (DPK/n2) + (DPR/n3)} \tag{3}$$

As with the similarity weight described in Section 2.2, the measure provides a score somewhere between 0 and 1 for the comparison of key and response concepts with respect to a given ontology. If a concept is missing or spurious, BDM is not calculated since there is no MSCA. If the key and response concepts are identical, the score is 1 (as with Precision and Recall). Overall, in case of an ontological mismatch, this method provides an indication of how serious the error is, and weights it accordingly.

The BDM itself is not sufficient to evaluate our populated ontology, because we need to preserve the useful properties of the standard Precision and Recall scoring metric. Our APR metric (Augmented Precision and Recall) combines the traditional Precision and Recall with a cost-based component (namely the BDM). We thus combine the BDM scores for each instance in the corpus, to produce Augmented Precision, Recall and F-measure scores for the annotated corpus, calculated as follows:

$$AP = \frac{BDM}{n + Spurious} \qquad and \qquad AR = \frac{BDM}{n + Missing} \tag{4}$$

while F-measure is calculated from Augmented Precision and Recall as:

$$F - measure = \frac{AP * AR}{0.5 * (AP + AR)} \tag{5}$$

5.2. Experiments with OBIE evaluation

The BDM metric has been evaluated in various ways in order to compare it with other metrics for evaluation and to test scalability issues. For the evaluation, a semantically annotated corpus was created for use as a gold standard. This is known as the OntoNews corpus [37]. This semantically annotated corpus consists of 292 news articles from three news agencies (The Guardian, The Independent and The Financial Times), and covers the period of August to October, 2001. The articles belong to three general topics or domains of news gathering: International politics, UK politics and Business.

The ontology used in the generation of the ontological annotation process is the PROTON ontology[3], which has been created and used in the scope of the KIM platform[4] for semantic annotation, indexing, and retrieval [33]. The ontology consists of around 250 classes and 100 properties (such as partOf, locatedIn, hasMember and so on). PROTON has a number of important properties: it is domain-independent, and therefore suitable for the news domain, and it is modular (comprising both a top ontology and a more specific ontology).

The aim of the experiments carried out on the OntoNews corpus was, on the one hand, to evaluate a new learning algorithm for OBIE, and, on the other hand, to compare the different evaluation metrics (LA, flat traditional measure, and the BDM).

The OBIE algorithm learns a Perceptron classifier for each concept in the ontology. Perceptron [38] is a simple yet effective machine learning algorithm, which forms the basis of most on-line learning algorithms. Meanwhile, the algorithm tries to keep the difference between two classifiers proportional to the cost of their corresponding concepts in the ontology. In other words, the learning algorithm tries to classify an instance as correctly as it can. If it cannot classify the instance correctly, it then tries to classify it with

[3] http://proton.semanticweb.org

[4] http://www.ontotext.com/kim

another concept with the least cost associated with it relative to the correct concept. The algorithm is based on the Hieron, a large margin algorithm for hierarchical classification proposed in [39]. See [40] for details about the learning algorithm and experiments.

We experimentally compared the Hieron algorithm with the SVM learning algorithm (see e.g. [41]) for OBIE. The SVM is a state of the art algorithm for classification. [42] applied SVM with uneven margins, a variant of SVM, to the traditional information extraction problem and achieved state of the art results on several benchmarking corpora. In the application of SVM to OBIE, we learned one SVM classifier for each concept in the ontology separately and did not take into account the structure of the ontology. In other words, the SVM-based IE learning algorithm was a flat classification in which the structure of concepts in the ontology was ignored. In contrast, the Hieron algorithm for IE is based on hierarchical classification that exploits the structure of concepts.

As the OntoNews corpus consists of three parts (International politics, UK politics and Business), for each learning algorithm two parts were used as training data and another part as test data. Note that although the tripartition of the corpus indicates three distinct and topically homogeneous parts of the corpus, these parts are used as training and testing data for the comparison of different algorithms, and not their performance. For this purpose, semantic homogeneity does not play a role.

For each experiment we computed three F_1 values to measure the overall performance of the learning algorithm. One was the conventional micro-averaged F_1 in which a binary reward was assigned to each prediction of instance — the reward was 1 if the prediction was correct, and 0 otherwise. We call this flat_F_1 since it does not consider the structure of concepts in the ontology. The other two measures were based on the BDM and LA values, respectively, which both take into account the structure of the ontology.

	flat_F_1	BDM_F_1	LA_F_1
SVM	73.5	74.5	74.5
Hieron	74.7	79.2	80.0

Table 1. Comparison of Hieron and SVM for OBIE

Table 1 presents the experimental results for comparing the two learning algorithms SVM and Hieron. We used three measures: conventional micro-averaged flat_F_1 (%), and the two ontology-sensitive augmented F_1 (%) based respectively on the BDM and LA, BDM_F_1 and LA_F_1. In this experiment, the International-Politics part of the OntoNews corpus was used as the test set, and the other two parts as the training set.

Both the BDM_F_1 and LA_F_1 are higher than the flat_F_1 for the two algorithms, reflecting the fact that the latter only counts the correct classifications, while the former two not only count the correct classifications but also the incorrect ones. However, the difference for Hieron is more significant than that for SVM, demonstrating an important difference between the two methods — the SVM based method just tried to learn a classifier for one concept as well as possible, while the Hieron based method not only learned a good classifier for each individual concept but also took into account the relations between the concepts in the ontology during the learning.

In terms of the conventional flat_F_1, the Hieron algorithm performed slightly better than the SVM. However, if the results are measured by using the ontology-sensitive measure BDM_F_1 or LA_F_1, we can see that Hieron performed significantly better than

SVM. Clearly, the ontology-sensitive measures such as the BDM_F_1 and LA_F_1 are more suitable than the conventional flat_F_1 to measure the performance of an ontology-dependent learning algorithm such as Hieron.

In order to analyse the difference between the three measures, Table 2 presents some examples of entities predicted incorrectly by the Hieron based learning system, their key labels, and the similarity between the key label and predicted label measured respectively by the BDM and the LA. Note that in all cases, the flat measure produces a score of 0, since it is not an exact match.

No.	Entity	Predicted label	Key label	BDM	LA
1	Sochi	Location	City	0.724	1.000
2	Federal Bureau of Investigation	Organization	GovernmentOrganization	0.959	1.000
3	al-Jazeera	Organization	TVCompany	0.783	1.000
4	Islamic Jihad	Company	ReligiousOrganization	0.816	0.556
5	Brazil	Object	Country	0.587	1.000
6	Senate	Company	PoliticalEntity	0.826	0.556
7	Kelly Ripa	Man	Person	0.690	0.667

Table 2. Examples of entities misclassified by the Hieron based system

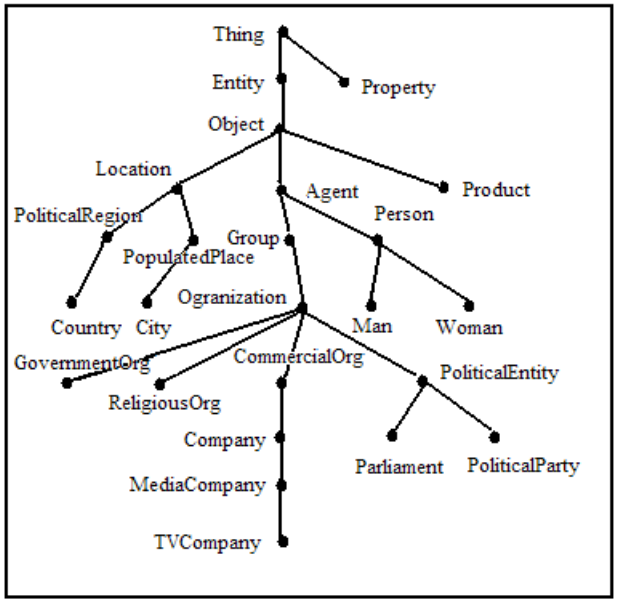

Figure 4. Subset of the PROTON ontology

All the concepts and their relations involved in Table 2 are illustrated in Figure 4, which presents a part of the PROTON ontology. This ontology section starts with the root node *Thing*, and has 10 levels of concepts with *TVCompany* as the lowest level concept. Note that the graph does not show all the child concepts for most of the nodes presented.

The conventional flat measure assigned each case a zero similarity because the examples were misclassified and the measure does not consider the structure of labels. On the other hand, both the LA and BDM take into account the structure of labels and measure the degree of a misclassification based on its position in the ontology. Hence they assign a non-zero value to a misclassification in most cases. Note that zero would be assigned in the case where the MSCA is the root node. In our experiments, all the concepts used were below the node "Entity" and so we used its immediate upper node "Thing" as root[5]. This meant that CP (the depth of the MSCA) was always at least 1, and hence there is no zero value for BDM or LA in our experiments. This is because we consider that if an entity's instance is recognised but with the wrong type, the system should have a non-zero reward because it at least recognised the instance in the first place. However, this could be changed according to the user's preference.

However, BDM and LA adopt different mechanisms in consideration of the ontology structure. In particular, the LA assigns the maximal value 1 if the predicted label is an ancestor concept of the key label, regardless of how far apart the two labels are within the ontological chain. In contrast, the BDM takes into account the similarity of two concepts in the ontology and assigns a distance-dependent value. The difference is demonstrated by the examples in the table. For example, in the Proton ontology, the predicted label *Organization* is the parent concept of the key label *GovernmentOrganization* in the second example, and in the third example the same predicted label *Organization* is 4 concepts away from the key label *TVCompany*. Hence, the BDM value of the second example is higher than the BDM value of the third example. In the first example, the predicted label *Location* is 3 concepts away from the key label *City* but its BDM value is lower than the corresponding value in the third example, mainly because the concept *Location* occupies a higher position in the Proton ontology than the concept *Organization*. Similarity is thus lower because higher concepts are semantically more general, and therefore less informative.

Another difference between the BDM and LA is that the BDM considers the concept densities around the key concept and the response concept, but the LA does not. The difference can be shown by comparing the fourth and the sixth examples. They have the same predicted label *Company*, and their key labels *ReligiousOrganization* and *PoliticalEntity* are two sub-concepts of *Organization*. Therefore, the positions of the predicted and key labels in the two examples are very similar and hence their LA values are the same. However, their BDM values are different — the BDM value of the fourth example is a bit lower than the BDM value of the sixth example. This is because the concept *PoliticalEntity* in the sixth example has two child nodes but the concept *ReligiousOrganization* in the fourth example has no child node, resulting in different averaged lengths of chains coming through the two concepts.

The BDM value in the fifth example is the lowest among the examples, mainly because the concept *Object* is in the highest position in the ontology among the examples. These differences in BDM scores show the effects of the adoption of chain density and branching factor as penalty weights in the computation of the score. These reflect the level of difficulty associated with the selection of a particular ontlogical class relative to the size of the set of candidates.

[5] "Thing" subsumes both "Entity" and "Property"

5.3. Discussion and Future work

The initial observation from our experiments is that binary decisions are not good enough for ontology evaluation, when hierarchies are involved. We propose an Augmented Precision and Recall measure that takes into account the ontological distance of the response to the position of the key concepts in the hierarchy. For this purpose we have developed an extended variant of Hahn's Learning Accuracy measure, called Balanced Distance Metric, and integrated this with a standard Precision and Recall metric. We have performed evaluations of these three metrics based on a gold standard corpus of news texts annotated according to the PROTON ontology, and conclude that both the BDM and LA metrics are more useful when evaluating information extraction based on a hierarchical rather than a flat structure. Furthermore, the BDM appears to perform better than the LA in that it reflects a better error analysis in certain situations.

Although the BDM gives an intuitively plausible score for semantic similarity on many occasions, it can be argued that in some cases it does not correlate well with human judgement. Examples 4 and 6 in Table 2 show counter-intuitively high similarity values for combinations of key and wrongly predicted labels, particularly in comparison with example 7. Note that as mentioned earlier, they are still better than the LA in that they distinguish different values for the two examples. From a human perspective, they also seem more wrong than the erroneous classification in Example 7, and slightly more wrong than those in examples 1 and 3. This indicates a need for further tuning the BDM score with additional cost-based metrics, in order to meet human judgement criteria. In such cases, this could entail the integration of a rule which boosts similarity scores for concepts within the same ontological chain (in a more subtle way than LA), and which lowers the score for concept pairs that occur in different chains. Work will continue on further experiments with the integration of such rules, including assessment of the correlation between BDM scores and human intuition.

6. Conclusion

In this chapter we have investigated NLP techniques for term extraction and ontology population, using a combination of rule-based approaches and machine learning. Starting from an existing method we developed for term recognition using contextual information to bootstrap learning, we have shown how such techniques can be adapted to the wider task of information extraction. Term recognition and information extraction, while quite similar tasks in many ways, are generally performed using very different techniques. While term recognition generally uses primarily statistical techniques, usually combined with basic linguistic information in the form of part-of-speech tags, information extraction is usually performed with either a rule-based approach or machine learning, or a combination of the two. However, the contextual information used in the TRUCKS system for term recognition can play an important role in the development of a rule-based system for ontology-based information extraction, as shown by the development of the GATE tools described in this chapter. Furthermore, the similarity metric used in TRUCKS to determine a semantic weight for terms forms the basis for a new evaluation metric for information extraction (BDM), which uses similarity between the key and response instances in an ontology to determine the correctness of the extraction.

Experiments with this metric have shown very promising results and clearly demonstrate a better evaluation technique than the Precision and Recall metrics used for traditional (non-ontology-based) information extraction applications.

Acknowledgements

This work has been partially supported by the EU projects KnowledgeWeb (IST-2004-507482), SEKT (IST-2004-506826) and NeOn (IST-2004-27595).

References

[1] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[2] D.G. Maynard and S. Ananiadou. Identifying terms by their family and friends. In *Proc. of 18th International Conference on Computational Linguistics (COLING)*, Saarbrücken, Germany, 2000.

[3] K.T. Frantzi and S. Ananiadou. The C-Value/NC-Value domain independent method for multi-word term extraction. *Journal of Natural Language Processing*, 6(3):145–179, 1999.

[4] D. G. Maynard. *Term Recognition Using Combined Knowledge Sources*. PhD thesis, Mamchester Metropolitan University, UK, 2000.

[5] G. Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, 1994.

[6] D.G. Maynard and S. Ananiadou. Term sense disambiguation using a domain-specific thesaurus. In *Proc. of 1st International Conference on Language Resources and Evaluation (LREC)*, pages 681–687, Granada, Spain, 1998.

[7] D. Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proc. of 14th International Conference on Computational Linguistics (COLING)*, pages 977–981, Nantes, France, 1992.

[8] S.J. Nelson, N.E. Olson, L. Fuller, M.S. Tuttle, W.G. Cole, and D.D. Sherertz. Identifying concepts in medical knowledge. In *Proc. of 8th World Congress on Medical Informatics (MEDINFO)*, pages 33–36, 1995.

[9] NLM. Unified Medical Language System (UMLS). Technical report, National Library of Medicine, http://www.nlm.nih.gov/research/umls/umlsmain.html.

[10] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal, Canada, 1995.

[11] Gang Zhao. *Analogical Translator: Experience-Guided Transfer in Machine Translation*. PhD thesis, Dept. of Language Engineering, UMIST, Manchester, England, 1996.

[12] E. Sumita and H. Iida. Experiments and prospects of example-based machine translation. In *Proc. of 29th Annual Meeting of the Association for Computational Linguistics*, pages 185–192, Berkeley, California, 1991.

[13] G. Rigau, J. Atserias, and E. Agirre. Combining unsupervised lexical knowledge methods for word sense disambiguation. In *Proc. of ACL/EACL*, pages 48–55, Madrid, Spain, 1997.

[14] A. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proc. of 19th International Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, 1996.

[15] J-Y. Magadur and G. Tabuteau. Semantic disambiguation in an information retrieval system. In *NLP+IA 96*, pages 148–154, Moncton, Canada, 1996.

[16] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

[17] D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274, 2002.

[18] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. *The GATE User Guide*. `http://gate.ac.uk/`, 2002.

[19] D. Maynard and H. Cunningham. Multilingual Adaptations of a Reusable Information Extraction Tool. In *Proceedings of the Demo Sessions of EACL'03*, Budapest, Hungary, 2003.

[20] D. Maynard, V. Tablan, K. Bontcheva, and H. Cunningham. Rapid customisation of an Information Extraction system for surprise languages. *Special issue of ACM Transactions on Asian Language Information Processing: Rapid Development of Language Capabilities: The Surprise Languages*, 2003.

[21] D. Maynard. Multi-source and multilingual information extraction. *Expert Update*, 2003.

[22] A. Mikheev, M. Moens, and C. Grover. Named Entity Recognition without Gazetteers. In *Proceedings of EACL*. Bergen, Norway, 1999.

[23] D. Maynard, K. Bontcheva, and H. Cunningham. Automatic Language-Independent Induction of Gazetteer Lists. In *Proceedings of 4th Language Resources and Evaluation Conference (LREC'04)*, 2004.

[24] N. Aswani, V. Tablan, K. Bontcheva, and H. Cunningham. Indexing and Querying Linguistic Metadata and Document Content. In *Proceedings of Fifth International Conference on Recent Advances in Natural Language Processing (RANLP2005)*, Borovets, Bulgaria, 2005.

[25] P. Kogut and W. Holmes. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. In *First International Conference on Knowledge Capture (K-CAP 2001), Workshop on Knowledge Markup and Semantic Annotation*, Victoria, B.C., 2001.

[26] F. Ciravegna and Y. Wilks. Designing Adaptive Information Extraction for the Semantic Web in Amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*. IOS Press, Amsterdam, 2003.

[27] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 358–372, Siguenza, Spain, 2002.

[28] J. Domingue, M. Dzbor, and E. Motta. Magpie: Supporting Browsing and Navigation on the Semantic Web. In N. Nunes and C. Rich, editors, *Proceedings ACM Conference on Intelligent User Interfaces (IUI)*, pages 191–197, 2004.

[29] S. Dill, J. A. Tomlin, J. Y. Zien, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, and A. Tomkins. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the* 12^{th} *International Conference on World Wide Web (WWW2003)*, pages 178–186, Budapest, Hungary, May 2003.

[30] P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web. In *Proceedings of WWW'04*, 2004.

[31] M. Dzbor, E. Motta, and J. Domingue. Opening up magpie via semantic services. In *Proceedings of ISWC 2004*, Hiroshima, Japan, 2004.

[32] L. K. McDowell and M. Cafarella. Ontology-Driven Information Extraction with OntoSyphon. In *5th Internal Semantic Web Conference (ISWC'06)*. Springer, 2006.

[33] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing and retrieval. *Journal of Web Semantics, ISWC 2003 Special Issue*, 1(2):671–680, 2004.

[34] D. Maynard. Benchmarking ontology-based annotation tools for the semantic web. In *UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology"*, Nottingham, UK, 2005.

[35] E. Agirre and G. Rigau. Word sense disambiguation using conceptual density. In *Proc. of 16th International Conference on Computational Linguistics*, volume 1, pages 16–23, Copenhagen, Denmark, 1996.

[36] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *Proc. of 15th National Conference on Artificial Intelligence (AAAI-98)*, pages 524–531, Menlo Park, CA, 1998. MIT Press.

[37] W. Peters, N. Aswani, K. Bontcheva, and H. Cunningham. Quantitative Evaluation Tools and Corpora v1. Technical report, SEKT project deliverable D2.5.1, 2005.

[38] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[39] O. Dekel, J. Keshet, and Y. Singer. Large Margin Hierarchical Classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML-2004)*, Canada, 2004.

[40] Y. Li, K. Bontcheva, and H. Cunningham. Perceptron-like learning for ontology based information extraction. Technical report, University of Sheffield, Sheffield, UK, 2006.

[41] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[42] Y. Li, K. Bontcheva, and H. Cunningham. SVM Based Learning System For Information Extraction. In M. Niranjan J. Winkler and N. Lawerence, editors, *Deterministic and Statistical Methods in Machine Learning*, LNAI 3635, pages 319–339. Springer Verlag, 2005.