Authoring Technical Documents for Effective Retrieval

Jonathan Butters and Fabio Ciravegna

Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, S1 4DP, Sheffield, UK, email {j.butters f.ciravegna}@dcs.shef.ac.uk

Abstract. In this paper we outline the design considerations and application of a methodology to author technical documents in order to improve retrieval. Our approach is firmly aimed at large organizations where variations in terminology at personal, national and international scales often impede retrieval of relevant knowledge. We first present the difficulties in performing entity extraction in technical domains and the role variation in terminology has in the information extraction task before outlining and evaluating a methodology that allows for effective retrieval.

1. Introduction

Effective Knowledge Management (KM) within large organizations relies on the capability to *capture*, *locate* and *exchange* relevant knowledge in a *timely* manner, however knowledge work is often a time consuming and expensive process [1], to draw conclusions from corpora requires the ability to accurately identify relevant knowledge within documents. To facilitate this, knowledge workers 1) *formulate* a search query – not necessarily using their own terms 2) *retrieve* relevant documents – in accordance with some metric, 3) *assimilate* information – assuming that their comprehension matches the document author's intentions, and 4) *assess* whether the knowledge is relevant to their situation – classify the document.

At each of these stages errors are introduced; systems only retrieve documents that contain the query string, real-world document recall is never 100%, comprehension of the document requires recognition of the concepts referred to by heterogeneous terms, and the degree of relevancy requires the identification of complex relations between entities. Large organizations are prone to the effects of *sublanguage* [2], where different groups of people use different subsets of language. In addition to this as people in technical domains are confronted with a large number of *specifications* (such as convergence towards a terminological standard based on geographical location e.g. *'hpc stage 5 shaft'* Vs. *'l5 drum, mod 31'*) and *standards* (including S1000D/ATA Spec 100/internal naming conventions e.g. *'72-31-53'* Vs. *'FK12345'* Vs *'HP Compressor stage 5 disc'*). The large number of valid terms,

along with the inevitable list of distortions and variations (word order, misspellings, morphological variations, e.t.c.) that arise during their use (Table 1) mean that terms vary both within and across documents and corpora. This further compounds the information retrieval task, as in order to increase recall, queries must be expanded to include a multitude of synonyms that may or may not exist within documents.

"Low Pressure Turbine Stage 2 Rotor Blade"
"LP2 Blade"
"FK42164"
"LPT 2 Blade"
"72-41-12"
"T800 LP Turbine Blade Stage 2"
"Turbine Blade"
"72-41-12-400"
"Blade, Turb l2"
"Blade, LPT"
"TurbinneBladee"
"FK12548"

Table 1 Examples of typical term variation for an '*LP2 Turbine Blade*' concept

By accurately capturing information and relations at the point of data creation, knowledge can be accessed more precisely and be comprehended in the manner the author intended.

The most precise method of capturing data at the point of creation involves the annotation of entities and their relations by domain experts, this is a complex and expensive [3], low recall operation [4] and when performed manually proves to be a bottleneck [5]. Methods that support the user through the annotation process reduce the time taken to annotate documents [6] and can be adapted to new domains (adaptive IE systems [7]) but are limited by their capability to extract named entities (perform NER). Extraction based on shallow NLP methods [8] may not be directly applicable due to the *deviant grammars* sublanguages frequently used [2], and also as many entities can appear with little context or no context at all (as a search keyword or within a sparse table for example). The current trend for automatic Named Entity and Relation Extraction in *open world* domains make use of lexical resources such as WordNet [9], Wikipedia [10, 11, 12, 13] and Google [14]. These resources are less practical in more technical and restricted domains such as *Aerospace*, *Biomedical* and *Automotive* where coverage is sparse. It is for these reasons that, Named Entity and Relation Extraction efforts in restricted technical domains tend to involve supervised [15, 16] or semi-supervised [17] machine learning techniques coupled with bootstrapping algorithms [18] where manually extracted entities are used as seeds in order to learn generalized extraction rules which in turn identify further seeds. These approaches are significantly affected by the annotation bottleneck and variations in terminology; the annotation bottleneck governs the number (and quality) of annotated documents where-as variations in terminology reduce the number of examples per class and negatively affects the benefits of bootstrapping.

Other methods to capture data at the point of creation involve authoring documents as a form where each field is linked to an ontological concept [19]. Although this method mitigates the need for information extraction on a document

level, instances of concepts still need to be recognized – for example, the terms *'comp drum'* and *'hp shaft'* may be applied as the concept 'part_removed' within two different documents, however these terms are synonymous and should be recognized as such. The inclusion of free-text boxes where authors are able to insert paragraphs of text means information extraction is sometimes still necessary.

Terminology Recognition is the ability to recognize the semantic class[1] and concept[2] referred to given a term, allowing a dereferencable URI[3] to be assigned. Terms vary for many reasons: use of *sublanguage*, *morphology*, *acronyms*, *abbreviations*, *lexical differences*, *word order* and *misspellings* [2]. We performed experiments on approximately 40000 Aerospace jet engine component terms linked to Part numbers. The results show that the number of synonymous terms used to refer to a particular part number increases in accordance with a power law with the popularity of that part number (Figure 1), this means that for any given concept it is unlikely that a gazetteer list of all valid terms could be compiled *a priori*. Terminology recognition systems therefore need the capability to identify term forms that have never been encountered before. This allows authors own terms to be used at a *document* level and concept URIs at a *metadata* level.

Due to term variation, Information Retrieval (IR) is a more involved task. Basic keyword search usually only returns documents that contain the query term, shrewd users often reformulate [20] their query (sublanguage) to use more popular terms or term variations indicative of a corpora they wish to target. More sophisticated search systems can account for linguistic features such as *affixes*, *morphology* and simple *synonyms*. A more effective search solution is be able to retrieve documents based on *conceptual similarity* [21], in this regime a URI is generated from the query term (the query URI uniquely identifies the concept referenced to by the query term). The query URI is then compared against URIs generated for indexed entities. A match indicates conceptual similarity even though terms used may have a high string distance[4].

In this paper we present a terminology aware, ontology-driven methodology for extracting knowledge from free-text at the point of creation. We utilize Terminology Recognition in order to recognize ontological concepts within the document in real time as it is authored. Ontological relationships are suggested and displayed between relevant collocated concepts. Cross media resources and external information is drawn together and presented to the user in order to make the identification of correct URIs and relations as non-intrusive as possible. Positive and negative examples of concepts and relations are fed back in order to improve the entity and relation extraction task. Entities and relations are then recorded in an

---

[1] Semantic Class – The type of entity, for example; 'part', 'feature' or 'mechanism'.

[2] Concept – A particular instance of a semantic class. Concepts may be referenced to by many synonymous terms for example *'FK12345'*, *'HPC5 blade'*, *'HP Comp stg.5 blade'*.

[3] URI – Uniform Resource Identifier; a unique string of characters used to identify a resource. Allows knowledge workers access to concept across corpora no matter how the concept is represented in text.

[4] String Distance – String distance metrics cost the number of character edits required to transform one string into another. A high string distance indicates a pair of dissimilar strings.

unambiguous and machine-readable format and are directly useable by downstream processes such as search engine indexer.

2. Requirements

The primary objective of our methodology is to provide a support capability for technical domain document authors in order to facilitate more effective document retrieval. We do this by applying Terminology Recognition as the document is authored in order to identify entities, concepts and relations within free-text. If Terminology Recognition is able to identify both the entity[5] and concept[6] from a term, no further action is required from the author as the term can be uniquely represented with the concept's URI. If on the other hand Terminology Recognition identifies an entity but not the concept, the author is required to specify the concept in order for the term to be uniquely identified. Concepts may not be identified for two reasons:

1. The author underspecified the term – for example *'bolt'* as opposed to *'combustor case lower bolt'* – in which case the author should use a more descriptive term or manually apply the URI of the concept they are referring to.
2. Terminology Recognition fails to classify the concept's URI, in which case Terminology Recognition requires further training.

The complexity in the entity, concept and relation identification task stems from the large number of ways in which entities, concepts and relations can be expressed. However, by recognizing variations in terminology, sublanguage may be used freely within the documents without compromising the ability to retrieve the document.

We will now present some design considerations and requirements for a system that is able to identify entities and relations by recognizing the terminology used.

Entity Extraction: The primary requirement is the ability to identify entities and recognize term variation. This differs from a typical Named Entity Recognition (NER) task as it not only entails the identification of semantic type (i.e. Date/Person/Organization/etc) but also requires the identification of the concept (i.e. *"McDonalds" = "MCD" = "Mac-Donnallds"*[sic] = *"Golden Arches"*). This allows the assignation of a URI that facilitates the identification of the concept across corpora and databases.

Resources: The poor coverage of technical domains within lexical resources such as WordNet, Wikipedia and Google means that these resources cannot be used for entity, concept or relation extraction. Pre-existing resources such as gazetteer lists, parts catalogues and taxonomic breakdowns should be leveraged in order to construct domain ontologies that provide basic relations such as holonym, meronym and hypernyms as well as domain specific concepts and relations.

Provide Feedback: If a concept can be uniquely identified (i.e. a URI can be assigned), no further user interaction is required. On the other hand, if no unique concept can be identified, the entity's term is either ambiguous (applicable to multiple

[5] e.g. 'Part' | 'Feature' | 'Date'

[6] e.g. 'Part Number 123' | 'Standard Feature # 12' | '1997-07-16T19:20:30+01:00'

concepts), may not reference a concept in the ontology, or Terminology Recognition may have failed to apply a URI. In other words, further action is required to identify the concept. The user should be alerted that the concept cannot be uniquely identified and that further action is required.

Information regarding relations should also be fed back to the user, when an entity is extracted, Terminology Recognition identifies collocated entities with which a relation may exist (in accordance with the domain ontology) and classifies the relation as appropriate. This must be displayed to the user.

Real Time: The approach must process documents in real time; entities and their relations should be extracted as the author types.

Large Scale and Maintainable: The approach must be maintainable, cost effective and scalable across large organisations. As the number of term variations increases with the number of authors in accordance to a power law (Figure 1) the need to decentralize the term management task becomes apparent. Devolving the term management task to document authors themselves allows a greater number of morphological differences, acronyms, abbreviations, lexical diversity, misspellings and differences in word order to be identified, this information can in turn be analyzed and used to better identify further term variations. By using the approach across a large organization it would be possible to conflate different sublanguages, this can be used to improve information extraction from legacy documents where terminology variation is a major difficulty when performing entity recognition.

Non-intrusive: The approach must be fast and non-intrusive, and operate in a similar manner to a spell-checker. In order to reduce the 'annotation bottleneck' the time taken to assign URIs to concepts must be kept to a minimum, this should be achieved by reducing the annotation task to a correction task wherever possible.

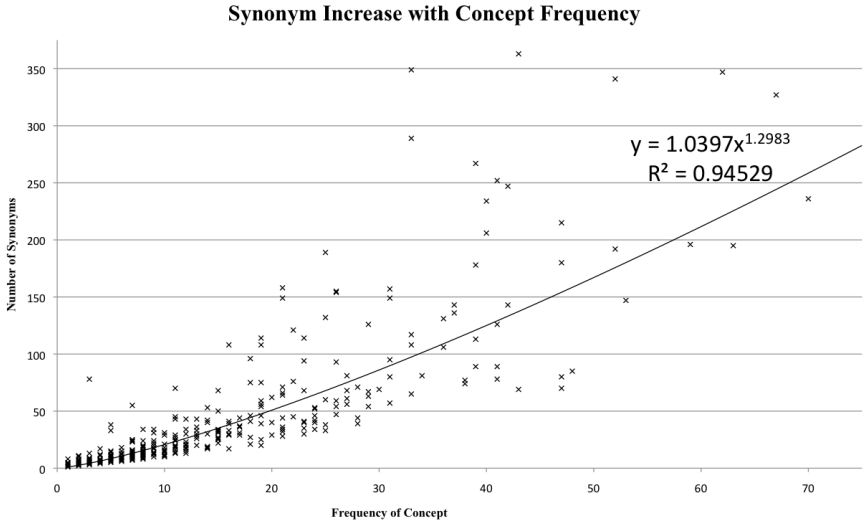

Figure 1 Number of synonyms increase with the frequency of concept occurrence in accordance with a power law

3. Resources

3.1 Ontology

We developed an aerospace jet engine domain ontology sourcing a large amount of information from the Rolls-Royce official engine parts catalogue. The ontology describes the relations between components (such as *meronyms* and *holonyms*) as well as providing *hypernyms*. We manually added concepts and relations for:

- Features – A feature is a location on a component. Components may have multiple features e.g. a blade has a *'leading edge'*, a *'face'* and a *'root'*. Although the number of aerospace jet engine features is finite, comprehensive component-feature lists do not currently exist.
- Deterioration Mechanisms – A deterioration mechanism is a process that causes a component or feature's condition to deteriorate. Examples are *crack*, *rub* and *dent*.
- Service Bulletins – A Service Bulletin is a type of report that discusses the condition of one or more components. Given any particular component it is useful for engineers to be able to retrieve relevant service bulletins in order to identify common faults. Service Bulletins are identified by their report number.
- Technical Variances – A Technical Variance is a type of report issued when an in-use component is found to be outside of design tolerances. Rolls-Royce produce the report which advises whether the component is safe to fly or not. Technical Variances are identified by their report number.

The ATA100 Spec[7] is an aerospace industry standard numbering specification used by aviation manufacturers, suppliers and airlines to identify aircraft systems. Rolls-Royce has extended this numbering specification to identify individual components within jet engines. The virtue of using the extended ATA100 Spec to refer to components rather than part numbers is that as components are improved/redesigned they are given a new part number (even though the component fulfills the same task), whereas the ATA number will remain the same, for this reason we use the R-R extended ATA100 Spec to form component URIs.

3.2 Terminology Recognition

We have developed a Terminology Recognition (TR) system that is in the process of being patented by Rolls-Royce (unfortunately therefore we cannot disseminate technical details yet). Terminology Recognition builds on the functionality of typical Named Entity Recognition (NER) system by identifying term

[7] http://en.wikipedia.org/wiki/Air_Transport_Association

variations such as acronyms, abbreviations, lexical, morphological, and syntactic differences.

Terminology Recognition does not require context in order to extract entities from text. This is because TR was designed to extract entities in different situations, in some cases there will be very little surrounding text (in a search query scenario for example, where there may only be one entity), in other cases there may be many collocated entities (a paragraph of text or an entire document). When context is present, TR makes use of surrounding terms in order to disambiguate underspecified and ambiguous concepts in order to apply a URI. When applied to text, TR outputs a list of entities associated with URIs along with entity offset information. If no single URI can be assigned to a given entity, a list of URIs is provided along with a probability for each. Each semantic class of entity TR extracts is linked to a concept within the domain ontology.

To be unambiguous, an aircraft engine component term needs to include: 1) an object (i.e. hypernym), 2) the type of object, and 3) the system the object is attached to. Several terms require positional modifiers on systems and objects (such as *'Stage 4 Compressor'* or *'rear case'*). To illustrate this, the term "High Pressure Compressor speed sensor", refers to a 'sensor' object with type 'speed' (i.e. the sensor senses speed), and the 'speed sensor' is attached to the 'HPC' system. Terminology Recognition models terms using its own ontological representation (term ontology), the relationships between the systems, types and positions are identified such that if one is missing, Terminology Recognition is capable of identifying why a term is underspecified.

4. User Interaction

As a document is authored, entities and relations are extracted and presented to the user, mal-extracted entities, concepts and relations are corrected by the author who ensures they match their intention. Through this interaction: 1) documents are written using the author's sublanguage (i.e. without forcing the author to use alternate terms) while allowing the retrieval of the document based on concepts rather than terms, and 2) provides training data in order to better classify mal-extracted entities, concepts and relations.

The interaction scheme is depicted in Figure 2 and is described in detail in the following sections.

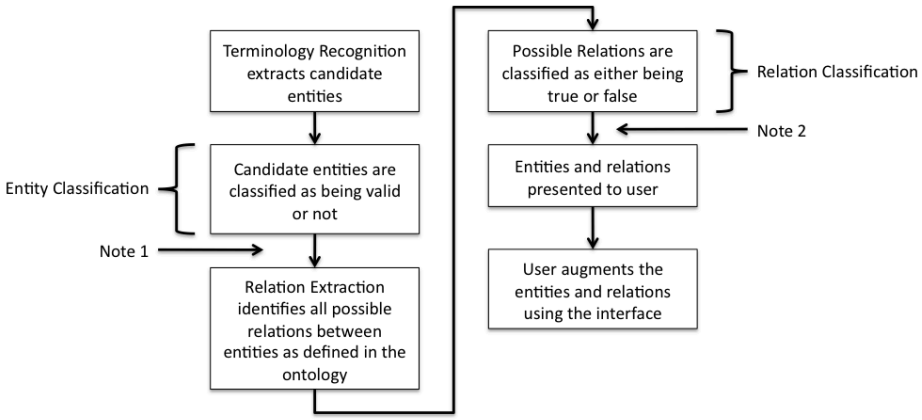

Figure 2 Interaction scheme

4.1 Knowledge Capture

We developed a prototype interface utilizing Terminology Recognition in the process depicted in Figure 2.

A. Entities

As a document is authored, Terminology Recognition (TR) extracts candidate entities, as further context is added TR is able to better identify whether a candidate entity is an instance of an ontological concept or not. For example, the term *'man'* is used in some contexts as a reference to 'EGCC[8]' the term may also occur freely in text with other senses. In order to classify entities based on context TR uses an SVM classifier with features including words immediately surrounding the entity along with information about the candidate entity itself such as canonical form and semantic type.

If the entity persists once context is taken into consideration a second classifier attempts to assign a URI. This classifier returns a list of possible URIs along with a confidence value for each. The confidence values are used to determine whether a URI should be automatically assigned or if the author should be notified that further action is required to disambiguate the entity. When a URI is assigned, the entity is uniquely identified, allowing the system to establish a link to references of the concept within other databases. In order to assist the author in choosing the correct URI (ATA100 code) the system queries the Rolls-Royce illustrated parts catalogue to display an image of the component referenced by the URI. Other databases that can be drawn together by the ATA100 code include list of materials and list of common faults, this information is displayed within a tool-tip when the author hovers over an

[8] EGCC – The International Civil Aviation Organization (ICAO) airport code for Manchester Airport, UK.

entity. The system notifies the user that an entity has been uniquely identified by using a distinct text color based on the entity type.

If TR identifies an entity but does not assign a URI (as confidence values are too low), the entity text is highlighted with the entity type color, this indicates to the user that further interaction is required in order to uniquely identify the concept. This may be remedied by either modifying the entity text to include a more detailed description, or manually selecting one of the lower confidence URIs. Underspecified entities are a common issue within aerospace domain technical documents. For example, there are many blades within an aerospace gas turbine, blades are situated within six different systems, three of these systems have multiple stages (up to a total of eight). The term *'IPC Stage 3 blade'* is fairly descriptive as it indicates the system, and position (stage) of the blade, the highest probability URI classified by Terminology Recognition (http://www.k-now.co.uk/r-r.owl#72-32-32-170) scores a probability of 86.53% (in this case this URI is correct as it references the 3rd stage intermediate pressure compressor blade concept). The second most probable URI scores a probability of 12.36% and references a component physically connected to the IPC stage 3 blade, the 3rd, 4th and 5th suggestion each score a probability <0.05%. The term *'Blade'* is clearly less descriptive, it does not indicate which blade is being referred to. The highest ranking classification suggests the Engine Fan Blades (http://www.k-now.co.uk/r-r.owl#72-32-32-170) - the largest and most prominent blades on the engine - with a confidence of 40.16%, the 2nd, 3rd 4th and 5th suggestions are various other blades in different positions across different systems each with a confidence value of approximately 10%. As extra characters are added to the term Terminology Recognition reclassifies the term in real time allowing the user to see whether enough information has been added to uniquely identify the concept.

In technical domain documents endophoric references (typically anaphoric) are commonly made. For example, when a component is described in one paragraph and then discussed in more detail in the following. *"The outlet guide vane case appeared to be in a serviceable condition. </P>The case was then removed for an intrascope inspection procedure"*. It is from situations like these that most under specified terms (and therefore low confidence URIs) are generated – This situation can be mitigated by resolving co-references. It is also important to recognize that the two entities 'outlet guide vane' and 'case' are co-referential as two sets of relations could be assigned to the entities, when it is more accurate to realize all relations occur on the same entity. Terminology Recognition attempts to establish ontological 'same_as' relations between co-referential concepts. Authors can manually establish and break same_as relations by right clicking an entity and selecting the 'Resolve Coreference' submenu, other entities with a same semantic type are suggested as possible co-references (both anaphoric and cataphoric within a window of 300 characters).

B. Relations

As the author continues to create the document, Terminology Recognition identifies pairs of entities between which the domain ontology specifies a relation may exist. These relations indicated to the user as arcs dawn within the interface connecting the subject and object entities. The arcs appear when the user hovers over

either subject or object entity. As each relation is represented with a <subject> <predicate> <object> triple, it is therefore possible to construct a sentence fragment (e.g. "*combustor case* has_damage *crack*") for each relation. These fragments are displayed within the tooltip as the user hovers over an entity, and provide an easy to digest test.

Terminology Recognition classifies all possible relations as either positive or negative using an SVM based classifier. The classifier uses features including the distance and words (tokenized on white space) between the subject and object entity, the relation predicate, context words surrounding the subject and object entity, as well details about the two entities themselves, such as any classified URIs. Positively classified relation is indicated with a more pronounced arc where as negative relations become faded.

Relations between entities may also be specified by the user, when an entity is selected, relations are read from the ontology. Appropriate entities from within a window of text (collocated within 300 characters of the entity) are suggested, the cardinality of the predicate within the ontology determines whether the relation may be established between multiple entities.

Within technical documents, the least ambiguous entities include part numbers, serial numbers, ATA 100 codes, Technical Variance numbers and Service Bulletin numbers. As these entities usually follow a strict syntax they can be identified with very high precision and recall. Part numbers can be used to identify components within other data sources (such as a parts catalogue or bill of materials), this information can then be used to accurately identify hypernyms, holonyms and meronyms. It is likely that an entity collocated with a reference number will share the same concept and Terminology Recognition uses this information as features in order to better classify entity URIs.

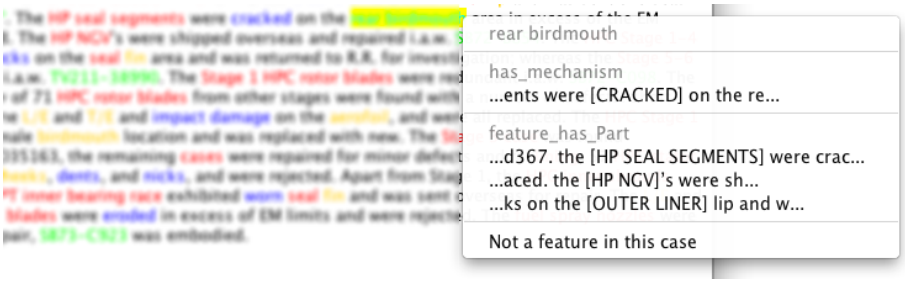

Figure 3 The interface. Author is establishing relations for the concept 'rear birdmouth' (Text has been blurred due to data confidentiality)

4.2 Learning

Normal author interaction is leveraged at every stage in order to provide training examples to improve entity extraction, coreference resolution and relation extraction. As the author corrects automatically classified entities and relations the features required for each task are recorded. The classifiers can them be retrained offline using both the existing and new training data.

5. Evaluation

We ran a series of experiments in order to assess how our methodology assisted authors at the document generation stage – in particular, we focused on evaluating the methodology's performance in supporting authors generate semantically rich documents.

The experiments are set in the Aerospace jet engine domain. As no gold standard annotated corpora for this domain exists, we divided our evaluation into a number of separate tasks each to evaluate separate processes of our methodology individually.

Rolls-Royce supplied two datasets; data set 1 comprised a collection of 88213 report summaries spanning three corpora. The summaries originated from the structured tables included within the header of each report, and so for each report summary we were able to accurately extract fields such as date, part number, ATA100 number, component term. Data set 2 comprised 4394 complete documents randomly selected from across 6 corpora.

We first assessed our methodology's ability to identify entities within free-text. As a document is authored entities must be identified, we evaluated Terminology Recognition (without the contextual classifier), *'Termex'* [22] and *'C-Value'* [23] ATR algorithms against TF-IDF as a baseline. In order to generate the statistical measures for the ATR algorithms we first considered each corpus individually, then split the documents 40% training, 60% testing, we used the training split to generate statistics for *domain pertinence*, *domain consensus* and *lexical cohesion* (Termex), *document frequency* (C-Value), and *inverse document frequency* (TF-IDF), before extracting entities from the test set. The ability to identify entities within the test corpus was measured in terms of precision and recall. In this experiment, *precision* represents the fraction of identified entities that were correct, if an entity was partially identified we counted this as a half correct extraction. *Recall* represents the fraction of entities that were identified by the algorithm.

	Corpus A			**Corpus B**		
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**
TF-IDF	12.00%	8.54%	9.98%	14.62%	7.32%	9.76%
Termex	41.69%	18.02%	25.16%	49.82%	21.30%	29.84%
C-Value	52.87%	34.86%	42.02%	62.40%	41.85%	50.10%
TR	69.03%	97.12%	80.70%	92.77%	98.30%	95.45%

	Corpus C			**Corpus D**		
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**
TF-IDF	16.33%	5.83%	8.59%	13.59%	6.21%	8.52%
Termex	41.34%	25.34	81.35%	51.43%	22.73%	31.53%
C-Value	60.29%	39.93%	48.04%	64.76%	43.86%	52.30%
TR	94.49%	98.10%	96.26%	85.14%	94.03%	89.36%

	Corpus E			Corp F		
	Pre	**Rec**	**F1**	**Pre**	**Rec**	**F1**
TF-IDF	12.99%	7.24%	9.30%	16.65%	11.02%	13.26%
Termex	43.56%	22.10%	29.32%	35.23%	31.54%	33.28%
C-Value	59.43%	41.11%	48.60%	53.36%	42.40%	47.25%
TR	94.32%	95.40%	94.86%	97.35%	98.21%	97.78%

Table 2 Precision and Recall across different corpora.

Our second experiment evaluated Terminology Recognition's ability to assign a correct ATA100 code (and therefore URI) given an entity term. Using data set 1 we first filtered out blank, malformed and invalid ATA100 codes using the official engine parts catalogue this resulted in 39034 high quality term-ATA pairs. TR's URI classifier was trained using a 40% training split and evaluated on the remaining terms. Terminology Recognition assigned the correct ATA with a precision of 87.64%.

Our third experiment evaluated the capability to learn and extract relations by measuring the precision and recall of automatically predicted relations at regular intervals (after 5, 10, 20, 40 and 80 documents). We manually asserted relations for the first five documents, then used the positive and negative training examples to train the classifier, we then evaluated the classifier while confirming correctly classified relations, asserting missed relations and removing incorrect relations in the following five documents, after which we retrained the relation classifier on the ten marked up documents. We continued until the classifier achieved a *recall* greater than 75%.

Relation Type	**Number of Training Documents**	**Precision**	**Recall**	**F-Measure**
Part_has_mechanism/ Mechanism_has_part	20	83.43%	78.35%	80.81%
Part_has_feature/ Feature_on_part	40	88.41%	75.20%	81.27%
SB_has_part	5	100%	94.21%	97.02%
TV_has_part	5	100%	98.40%	99.19%

Table 3 Number of documents required to achieve a recall > 75%

Our final experiment simulated the task of producing semantically rich technical documents. We evaluated the time taken to identify entities and relations within a sample of 20 documents from data set 2 using TR to identify entities and relations in contrast to manually identifying and asserting entities and relations.

		Precision	**Recall**	**F-measure**	**Annotator Agreement**
Entities	**TR**	98.74%	99.43%	99.08%	99.49%
	Manually	98.62%	99.01%	98.81%	84.37%
Relations	**TR**	96.21%	92.81%	94.48%	99.21%
	Manually	94.38%	91.34%	92.84%	92.37%

Table 4 Manual extraction Vs TR Assisted extraction

	Time taken per document (seconds)
TR assisted	494
Manually	1239
Average length of document = 1552 words.	

Table 5 Time taken to annotated documents manually and with TR assistance

Our experiments show that time taken to semantically mark up documents is significantly reduced when variations in terminology are identified, and that entity and relation extraction across corpora within a large organization can be improved through non-intrusive identification of entities. In our first experiment we showed that Terminology Recognition outperforms both the C-Value and Termex ATR algorithms in identifying entities across different corpora within the Aerospace domain. C-Value outperformed Termex across all corpora as it is more capable at extracting multi-word terms (typically component terms which occured frequently). The terms Termex identified tended to be single word feature, and mechanism entities. Once an entity text has been identified, Terminology Recognition can identify the concept and assign the correct URI 87.64% of the time.

Our third experiment demonstrated that for most relation types, very few training documents were required to achieve a reasonable precision and recall. The relation needing most documents to train are the symmetric relations part_has_feature and feature_has_part, this is mostly due to the amount of variation in how these relations are expressed within the documents, the relationships SB_has_part & TV_has_part required very few training examples due to the very regular way they are expressed. Our final experiment showed recognizing variation in terminology can dramatically reduce the amount of time taken to semantically enrich documents, we achieved a 60.13% reduction in the time taken to assign URIs and establish relations.

6. Discussion&/conclusions

Applying rich semantic information to documents allows documents to be retrieved and consumed in the manner the author intended. Unfortunately, the task of annotating technical documents is extremely complex and expensive and open domain solutions based on WordNet, Wikipedia and Google are not directly applicable due to poor coverage. Our experiments show that by recognizing variations in terminology, we can achieve a 60% decrease in the time taken to identify entities and relations in aerospace domain documents. In this paper we outlined requirements in order to author technical domain documents for effective retrieval, our methodology:

1. Identifies entities and relations within free text affording a semantically searchable ontological representation of the knowledge within the document. Identified entities and relations are indicated to the author allowing them to ensure the classification matches their intention.
2. Operates in real time, allowing authors to mark up their document as it is written.
3. Is applicable across different corpora within large organizations, and improves in performance as the system is used.

4. Operates in a similar manner to a spellchecker without disrupting the authoring process.

One concern that arises from our experiments is that as authors become more and more familiar with the automatically extracted entities and relations authors may begin to trust and accept Terminology Recognition's suggestions without examining whether they are correct or not and as these entities and relations are used as training examples the feedback provides further strength to the misclassification.

5. References

1. Just-in-Time Delivery Comes to Knowledge Management, Harvard Business Review, Vol. 80, No. 7, July 2002.
2. R. Kittredge and J. Lehrberger. (1982). Sublanguage: Studies of Language in Restricted Semantic Domains. de Gruyter
3. Engelson, S.P. and Dagan, I., 'Minimizing manual annotation cost in supervised training from corpora', in Proceedings of the 34th annual meeting on Association for Computational Linguistics (1996).
4. Wilson, T, et al Wiebe, J, and Hoffmann, Paul, 'Recognizing contextual polarity in phrase-level sentiment analysis', in 'Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing', 2005.
5. Schlueter, S, Dong Q, and Brendel V, 'GeneSeqer@PlantGDB: gene structure prediction in plant genomes', in 'Nucleic Acids Research, 2003, Vol. 31, No. 13 3597-3600, Oxford University Press', 2003
6. Grishman, R 'Adaptive Information Extraction and Sublanguage Analysis' In *Proceedings of IJCAI Workshop on Adaptive Text Extraction and Mining,* 77-79. *2001*.
7. Ciravegna F., Dingli A., Petrelli D., Wilks, Y. *User-System Cooperation in Document Annotation based on Information Extraction*. In Proc. of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), Springer. 2002.
8. Ciravegna, F. 'Adaptive information extraction from text by rule induction and generalisation', In '*Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)'*, 2001.
9. Culotta, A and Sorensen, J., 'Dependency tree kernels for relation extraction', In *Proceedings of 'the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)'*, 2004
10. Z. Zhang and J. Iria. A Novel Approach to Automatic Gazetteer Generation using Wikipedia.', In *Proceedings 'of the ACL'09 Workshop on Collaboratively', 2009*
11. S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In R. C. Moore, J. A. Bilmes, J. Chu-Carroll, and M. Sanderson, editors, HLT-NAACL. ACL, 2006.
12. M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In AAAI, pages 1419–1424. AAAI Press, 2006.

13. A.Toraland, R.Munoz, 'A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia.' In 'Workshop on New Text, 11th Conference of the European Chapter of the Association for Computational Linguistics', 2006.
14. Pantel, P and Pennacchiotti M,. 'Espresso: Leveraging generic patterns for automatically harvesting semantic relations'. In ACL, 2006.
15. R. Feldman, B. Rosenfeld, S. Soderland, and O. Etzioni. 2006. Self-supervised relation extraction from the web. In ISMIS, pages 755–764.
16. E. Agichtein. 2006. Confidence estimation methods for partially supervised relation extraction. In SDM 2006.
17. J. Chen, D.-H. Ji, C. L. Tan, and Z.-Y. Niu. 'Semi-supervised relation extraction with label propagation.', In 'HLT-NAACL', 2006.
18. E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP-2003*.
19. Bhagdev, R., Chakravarthy, A., Chapman, S., Ciravegna, F., Lanfranchi, V.: Creating and Using Organisational Semantic Webs in Large Networked Organisations. In: Proceedings of the 7th International Semantic Web Conference, Karlsruhe, Germany (October 2008)
20. Liu, H., Lieberman, H., Selker, T. 2002. GOOSE: A Goal-Oriented Search Engine With Commonsense. Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, (AH2002) Malaga, Spain.
21. F. Giunchiglia, U. Kharkevich, and I. Zaihrayeu. 'Concept search.' In Proc. of 'ESWC, pages 429–444', 2009.
22. Sclano, F., and Velardi, P. 'Termextractor: a web application to learn the shared terminology of emergent web communities.' In 'Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007).', 2007
23. Frantzi, K. T., and Ananiadou, S. 'The c/nc value domain independent method for multi-word term extraction.', in 'Journal of Natural Language Processing utilization in the information search and delivery system for IBM technical support. IBM Systems Journal 43(3):546–563.', 2003

Acknowledgements
This work was carried out in association with Rolls-Royce plc. Sponsored by the UK Engineering and Physical Sciences Research Council Case Studentship number 0800133X.