Concept Chaining Utilizing Meronyms in Text Characterization

Lori Watrous-deVersterre
Information Systems
New Jersey Institute of Technology
Newark, NJ 07003, USA
llw2@njit.edu

Chong Wang
Information Systems
New Jersey Institute of Technology
Newark, NJ 07003, USA
cs87@njit.edu

Min Song
Information Systems
New Jersey Institute of Technology
Newark, NJ 07003, USA
min.song@njit.edu

ABSTRACT
For most, the web is the first source to answer a question formulated by curiosity, need, or research reasons. This phenomenon is due to the internet's ubiquitous access, ease of use, and the extensive and ever expanding content. The problem is no longer the need to acquire content to encourage use, but to provide organizational tools to support content categorization that will facilitate improved access methods. This paper presents the results of a new text characterization algorithm that combines semantic and linguistic techniques utilizing domain-based ontology background knowledge. It explores the combination of meronym, synonym, and hypernym linguistic relationships to create a set of concept chains used to represent concepts found in a document. The experiments show improved accuracy over bag-of-words based term weighting methods and reveal characteristics of the meronym contribution to document representation.

Categories and Subject Descriptors
H.3.1 **[Information Storage and Retrieval]:** Content Analysis and Indexing – *dictionaries, indexing methods, linguistic processing;* H.3.7 [**Digital Libraries**]: Miscellaneous

General Terms
Algorithms, Performance, Reliability, Experimentation

Keywords
Concept extraction, text characterization, clustering, digital libraries, machine learning, natural language processing, ontology

1. INTRODUCTION
For most, the web is the first source to answer a question formulated by curiosity, need, or research reasons. This phenomenon is due to the internet's ubiquitous access, ease of use and the extensive amount of content. Internet access is available at home, work and via mobile connections just about everywhere else. Ease of use is due to the extensive indexing of content used by search engines. Finally, authoritative, amateur, or simply voyeurs into content generation provide a steady stream of new material for digestion. With this explosive growth in digital content comes the frustration of "weeding through" useless content that matches terms used in indexing. We need better text characterization representation to improve management and access methods used for digital content.

Conference'10, Month 1–2, 2010, City, State, Country.

Current management tools use indexing and cataloging techniques based on age-old methods used by libraries, museums, and archives. The structured information captured by these tools is accessed by end-users to find content but these predefined classifications may not capture user's information needs. Users of catalogues often spend time examining the digital content retrieved to determine the correct terminology to use. This translation process from what the user wants and how it is indexed is manual and time consuming. In recent years, the need for information access systems that can capture the cultural diversity of its users while providing structural information to support effective retrieval has been identified [1].

Cultural diversity influences the concepts of a document and the indexing/cataloguing of a document in a digital library. Ontologies capture some of this cultural diversity. In this paper, we use the background knowledge provided by ontologies to extract and chain together a document's major concepts. We also capture how much of a document is devoted to each concept chain, creating a representative concept signature.

While current research explores the incorporation of ontologies for concept generation, most only utilize synonyms to identify concepts and hypernyms to calculate their frequencies ([2], [5]). In this paper, we present a novel algorithm that uses the linguistic meronym relationship to chain together concepts that would otherwise show no relationship. This additional information about a document can improve text characterization and the quality (accuracy) of indexes in a digital library. Additionally, we explore a pruning technique to the final document representation that leaves the significant concepts and removes noise. The experiments show improved accuracy with meronyms included and reveal characteristics of the meronym contribution to document representation.

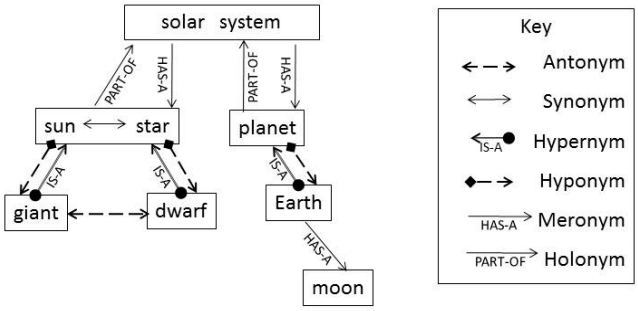
In this paper we provide a summary of related work (Section 2), followed by a description of the concept chain algorithm (Section 3). Section 4 describes the experimental results obtained with comments on the methods and limitations. We conclude in Section 5 with suggested future directions for our research.

2. BACKGROUND AND RELATED WORK
The explosive growth in digital content emphasizes the need to develop automated management (organizational) and access (discovery) tools to support the processing of digital content for information access systems. Organization of this generally unstructured content requires one to identify the scope, concepts, and purpose of the resource and then analyze the relationships of

the concepts to provide an overall understanding of the document [7]. A concept is a set of words that have semantic relationships [2]. Representing a document as a set of concepts provides a richer representation which, when used with clustering techniques, makes the resulting index scheme more useful [2]. This paper suggests a content analysis method for the storage and retrieval of textual documents utilizing a set of concepts to represent a document. The novelty in this approach is in the set of relationships used to associate the concepts and the resulting concept signature created to represent the document.

In the concept chain algorithm, we use an ontology to provide the concepts and relationships. An ontology contains a shared, controlled vocabulary which models a specific domain with the definition of concepts and their properties and relations [8]. WordNet is an ontology popularly used in natural language processing. Using WordNet's background knowledge, text documents are analyzed for concepts based on relationships between terms. Common linguistic relationships are antonyms (opposite meaning), synonyms (similar meaning), hypernyms ("IS-A" generalization of a term), hyponyms (more specific meaning of a term), holonyms ("PART-OF" relationship), and meronyms ("HAS-A" relationship). These relationships are diagramed as a concept map and shown in Figure 1:

Figure 1. Concept map using natural language relationships



Our study examines hypernym, synonym, and meronym relationships. Hypernym relationships form a directional "IS-A" connection between two terms that moves from a specific meaning to a more generalized one ("Earth IS-A planet"). Many studies have been performed to automatically extract these relationships from unstructured text, such as in Snow et al. [4]. Unlike hypernyms, terms which are synonyms can replace each other and still hold a similar meaning. For example, "sunshine" and "sunlight" terms may be used interchangeably in a sentence without significant loss of meaning. Meronyms are a bit more complex. Girju et al. [9] defined six types of meronyms which WordNet consolidates three categories; *member-of* (faculty HAS-A professor), *stuff-of* (tree HAS-A wood), and *part-of* (solar system HAS-A sun). Additionally, Girju et al identifies the *part-of* category as the most prominently used while Miller [10] indicates meronym transitivity may be optional as one moves away from the original relationship. For example, "Earth HAS-A moon" but the "plant HAS-A moon" relationship is optional (not all planets have moons).

WordNet has been used in numerous document-clustering experiments. Some of the earliest uses of WordNet in text categorization supported techniques to address effectively the classification of low frequency categories [11]. Green, in [12], used WordNet's hypernym and hyponym links to build lexical chains to analyze the similarity between information in different paragraphs. Hotho et al. in [5] showed utilizing background knowledge (i.e., relationships) between terms improved document-clustering. Hung and Wermter in [13] present three text vector representations, two of which used hypernym as concepts to improve classification accuracy. Recupero in [14] builds on past research using WordNet's hypernym relationship to improve on vector representation and clustering. In [2], Zheng et al. used WordNet relationships with noun phrases to analyze clustering improvements. Wang and Taylor [15] used WordNet to capture hypernym relations in short text documents creating clusters of concepts called concept forests to represent a document. In [6], Elberrichi et al, used WordNet to create a concept vector format they compared to traditional bag-of-word vector representation.

Except for [2], all these methods use single term analysis (using synonyms) and calculate term frequency from hypernyms. In fact, many of the papers listed suggest using more than one relationship as a future area research.

Accurately identifying concepts for categorization purposes is fraught with time-consuming manual analysis by content experts and librarians. A digital library catalog/index must represent the digital content and reflect the expectations of its users. Automating this process requires new techniques in concept extraction to analyze any size document and capture main concepts based on the appropriate domain. This paper describes extension to existing natural language and machine-learning techniques to improve the accuracy of extracting concepts from small text based resources and grouping them appropriately.

The selection of terms is a critical first step in concept generation. Terms with multiple meanings (polysemy) create ambiguity, while a term that is similar (synonyms) to others or have a degree of generalization (hypernym) can strengthen the importance of a concept. For these reasons, term frequency calculations often use hypernym and synonym information once ambiguity is resolved [7]. We also use this approach in our algorithm but the novelty of our approach is the inclusion of meronyms. The choice of meronyms stems from the idea of finding mechanisms to improve frequency measures for significant terms in short text documents without over constraining larger documents.

Some meronyms studies have been conducted as outlined by Yang and Callan [16]. Basu et al. in [17] developed a set of measures for different lexical relationships, including meronyms to identify the average semantic difference (i.e., the weight of an edge between two terms). Meronyms were given the same weight as hypernyms in this study. Berland and Charniak in [18] and Girju et al. in [9] suggest techniques for identifying meronyms for the specific use of incorporating them into taxonomies so they may be used in concept extraction. In [2], Zheng et al., used meronyms as the relationship to support clustering and found it to be not as good as hypernyms and holonyms. The novelty of our study examines the effects of weighing meronyms differently than synonyms or hypernyms when incorporating them into a frequency count for text characterization.

3. CONCEPT CHAIN ALGORITHM

The main steps in our concept chain algorithm include document preprocessing, concept/synset extraction, concept chain construction, and concept purification. Document data preprocessing is discussed in section 3.1. Section 3.2 discusses concept extraction based on WordNet's synsets. The last step that forms concept chains is described in section 3.3. Table 1 shows the complete concept chain algorithm.

Table 1. Concept Chain Algorithm.

Input: Text based digital resource

Output: Set of weighted concepts

Method:

Given a text based digital resource,

1. Identify Parts of Speech, retain nouns
2. Tokenize text and remove stopwords
3. Stem words (terms) and count frequencies
4. Choose the first synset of each word and discard others

   For each word A and its first synset S_A,

   Let P → S_A

   Do

   1. For each word B, other than A, and its first synset S_B,

      Do

      If S_B and all synsets in P are in the same hypernym (same branch), then do

      Add S_B into P

      Else if S_B and one synset in P are in the same synonym, then do

      Add S_B into P

      Else if S_B and all synsets in P are in the same meronym, then do

      Add S_B into P
   2. Add P into final concept chains
5. Compute SCR for each concept chain and rule out the concept chains whose SCR are lower than 3%

3.1 Data Preprocessing

The first step of our algorithm performs part-of-speech (POS) tagging for a given document and only retains nouns. After tokenizing each sentence, we convert all words to lowercase and filter stopwords. In our experiment, we use the NLTK stopwords corpus [19]. Finally, WordNet's morphology function stems the remaining words to find a possible base form for the given word. Using the given POS, the algorithm recursively strips affixes until a form in WordNet is found. As the text is processed, we record the frequencies of stemmed words in the document.

3.2 Concept/Synset Extraction

WordNet [10] groups English words into sets of synonyms called synsets. Synset words can be used interchangeably without significant change to the meaning or concept discussed in the document. In WordNet, synonyms are included in multiple synsets to represents the different meanings or concepts. Through a word's context, its meaning is derived but there may be ambiguity. For instance, the sentence "How much dough do you have?" has two meanings: a quantity of material used in cooking or the slang version to represent the concept "money". Identifying the correct meaning is the challenge of word sense disambiguation (WSD). In selecting a synset to represent a word's concept, we reduce complexity by eliminating the other word senses but in doing so we may sacrifice accuracy by selecting the wrong sense. In building the document's concept chains, we have to decide which synset represents the proper meaning of the word. Assigning a proper synset to each word is a form of WSD.

In [15], if a term has multiple senses, Wang et al. added only those senses that paired with other terms, making them candidate concept chains. This slightly reduced complexity while maintaining accuracy but we found this solution introduced noise preventing the capture of a concept signature for each document.

For example, the word "unit" has six senses in WordNet. It is used most frequently to mean a unit of measurement. This sense usually does not add uniqueness to a document's representation but "unit's" third sense defines an organization regarded as part of a larger social group. In this sense, a "team" and "crew" can create many IS-A relationships creating a more unique document signature by capturing the concept that discusses an organizational unit. But, if a document contains a high occurrence of "unit" as its first sense as well as many words that form IS-A relationships with the third sense, such as "team" and "crew", the merging result, {'unit', 'team'} and {'unit', 'crew'} are misleading and become noise.

Moreover, too many irrelevant candidate concept chains such as the two just described may dilute or over constrain the significant concepts. As discussed in the next section, the pruning of "noisy" concept chains may result in only a few or, worse, no concept chains remaining to represent the document. In our experiments when using all possible senses of a term we discovered about 20% of the documents processed became too constrained. This prevented the final generation of a unique set of concept chains to form a concept signature for each document.

To address this issue, we adopted the first synset of a word which are ordered by popularity in WordNet [10]. Adopting only the first sense reduces noise and total semantic content weight in the purification phase but it also reduces the potential accuracy of the correct word sense selected. In future plans, we will explore other WSD algorithms to improve accuracy.

3.3 Concept Chain Construction

Once synsets are determined, they are used to discover semantic relationships among words with the help of the WordNet ontology. Every pair of words in the document is checked to determine if their synsets in WordNet have a hypernym, synonym or meronym relationship. The set of concept chains and their proportionate contribution to the document is based on a weighted term frequency contained in a concept chain. The output of this phase is a set of candidate concept chains.

We define *a group of words are in same hypernym relationship (IS-A relationship), only when every pair of words in the group is in same hypernym branch in WordNet. They must form a chain without bifurcation.* Words in one hypernym branch are merged into a hypernym chain if they have the same relationship. Words with no hypernym relationships or are not directly connected, stay in distinct hypernym chains.

Let's consider the two scenarios shown in Figures 2 and 3. The line above "vehicle" indicates this is only a subset of an entire WordNet hypernym tree. In WordNet, "vehicle" and its several hyponyms follow the hierarchy shown. Each word represents the synset it contains.

Figure 2. IS-A relationship Scenario 1

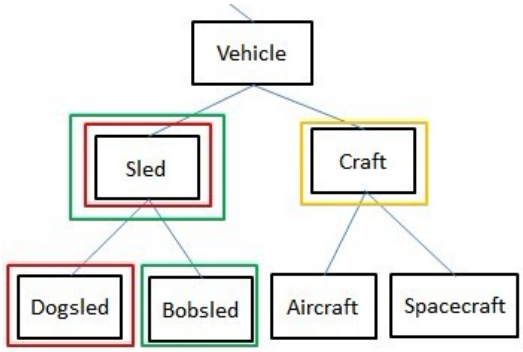


Figure 3. IS-A Relationship Scenario 2

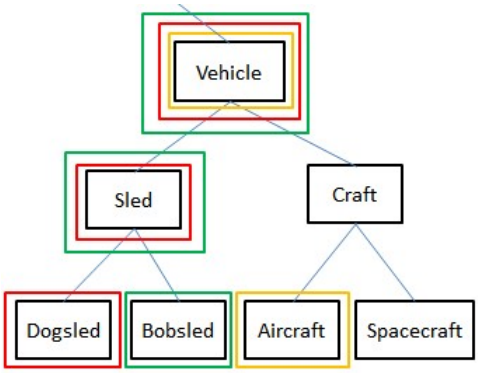


Figure 2 assumes a document contains "sled", "craft", "dogsled", and "bobsled". "Sled" and "dogsled" are grouped due to the IS-A relationship. Then "sled" and "bobsled" are grouped separately because of same reason but note that {'sled', 'dogsled'} and {'sled', 'bobsled'} are in different concept chains and that "dogsled" and "bobsled" are not in same hypernym branch due to our definition of a hypernym relationship. In this same figure, "craft" has no hypernym in the document so it is in a concept chain by itself. Thus, the output of concept chain construction will be {'craft'}, {'sled', 'dogsled'}, and {'sled', 'bobsled'}.

In Figure 3, a different document contains "dogsled", "vehicle", "aircraft", "sled" and "bobsled". After forming three concept chains: {'sled', 'dogsled'}, {'sled', 'bobsled'}, and {'aircraft'}, "vehicle" is detected. This is added into all three concept chains. Therefore, the output is {'sled', 'dogsled', 'vehicle'}, {'sled', 'bobsled', 'vehicle'}, and {'aircraft', 'vehicle'}. Note that {'aircraft', 'vehicle'} is valid even if "craft" is not in the document. We can set a distance threshold to limit the allowed maximum distance of two words in a branch. If this distance threshold is 1, then "vehicle" and "aircraft" will not be grouped because the distance between them is 2. In our project, we set the distance threshold to infinity so all concepts will be included. With this approach, we are able to capture multiple chains without over constraining the document.

In addition, we check synonyms to determine if they should be included in a concept chain. If two words in WordNet have the same first synset number, then they are synonyms. The concept chain algorithm examines a word's first synset. If it matches an existing word's first synset in a concept chain, then we add the word to the same concept chain. Note, a word may match multiple concept chains. When this happens, the concepts in these chains become more significant and reflect the major concepts in a document.

During concept chain construction we also include the three kinds of meronym relationships found in WordNet but we do not consider if a meronym relation is optional or mandatory – we include both. Simply, if several words are in same meronym tree in WordNet, they are added to one concept chain. For instance, a 'solar system' must have a 'sun' but 'planets' are optional. If 'solar system' and 'planet' are found in a document they will be added to the same concept chain.

3.4 Purification

The output of last step is a set of candidate concept chains for each document. However, as stated in the previous section, some concepts may appear as noise and misrepresent the significant semantic meaning of the document. These irrelevant candidate chains must be filtered. In order to do this we must quantify the relevancy of each concept chain and then provide a threshold for pruning.

Using Wang et al.'s [3] terminology, the semantic content weight (SCW) of a concept chain is the sum of the frequencies of all the words found in that specific concept chain. As indicated in earlier sections, *wordfreq* is a combination of synonym, hypernym, and meronym relationships. So, if the i^{th} concept chain has j words then,

$$SCW_i = C \cdot \sum_{j=1}^{j} wordfreq_j$$

Therefore, the sum of all SCWs is the total weight for the document. While calculating the SCW of a concept chain, a coefficient, *C*, is applied to the frequency from 0 and 1, where 0 excludes meronyms in the relationship, while a 1 provide a full weight equivalent to a IS-A relationship.

With this information we can represent each concept chain's SCW as a fraction or rate of the total document. Given n concept chains in a document, the semantic content rate (SCR) of the i^{th} concept chain can be described as:

$$SCR_i = \frac{SCW_i}{\sum_{n=1}^{n} SCW_n}$$

Once the SCR's for all concept chains have been established, a concept chain can be ruled out or kept depending on whether the quotient of its SCR and total SCR of the document is larger than a SCR threshold. In our experiment, we set SCR threshold to 3%.

Wang et al.'s work mapped stemmed words to particular synsetIDs. If several stemmed words mapped to the same synsetID, the word frequency value of this synsetID became the sum of the word frequency values of these associated words. This became the *wordfreq* used to compute SCW [15]. Although this approach recognizes synonyms and can support WSD it also loses specific information about a document which can lead to incorrectly cataloging a document.

For instance, 'Java' has multiple meanings ('island', 'coffee', 'programming language'). The " island" synset is the most popular usage of the word sense. Since our algorithm picks the most frequently used concept as part of the WSD process a document about 'Java programming' would be misclassified. Using just the synsetID in the final representation of the document would further exacerbate the problem. Instead, we maintain the more specific information, which is the document term, to support

clustering or classification methods to catalog/index it more accurately.

4. EVALUATION

To evaluate the accuracy of the representation of a document produced by the concept chain algorithm, we compared it to two common text classification techniques: vector space model (VSM) and term frequency-inverse document frequency (TF-IDF). For each classification technique, we converted the documents in a dataset to the appropriate representation. In VSM and TF-IDF, stopwords were removed but the rest of the words were not stemmed by WordNet's morphology function. We then performed a clustering algorithm to group the documents. The results were analyzed against the categories specified in the collection to determine accuracy. Accuracy is a measure of how many documents correctly clustered over the entire number of documents in the test set:

$$Accuracy = \frac{Total\ number\ of\ docs\ correctly\ clustered}{Total\ number\ of\ documents}$$

4.1 Similarity Computations

To compare the concept chain algorithm, we need to develop semantic vectors based on concept chains to represent semantic features of these documents. A document vector consists of attributes, each of which represents a stemmed word. The values of these attributes are the word frequencies. Taking Figure 2 as an example, its output of concept chain construction is {{'craft'}, {'sled', 'dogsled'}, {'sled', 'bobsled'}}. Making up the word frequencies, the semantic vector of this document is:

craft,	sled,	dogsled,	bobsled
{3,	2,	3,	4 }

Note, the word frequency of 'sled' can potentially be higher than the count in the document since it is found in two sub-chains; {'sled', 'dogsled'}, {'sled', 'bobsled'}. We use this vector to compute similarity between this document and others. Similarity measures are used to group similar document together into distinct clusters based on feature that is being measured against. This is a common means in text processing used to classify documents. Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. The closer the similarity value reaches 1, the more similar the documents:

$$\text{Similarity} = \cos\theta = \frac{A \cdot B}{||A||\,||B||}$$

We use this same measure when computing the similarity between the text representations used by VSM and TF-IDF. VSM is an algebraic model to represent text documents. It treats a document as "a bag of word" and represents it as a vector [20]. TF-IDF measures the importance of a word in a document based on the number of times the word appears in the document but is offset by the frequency of the word in the corpus [21]. Since VSM and TF-IDF are very common and widely accepted models to represent documents, we choose them as baseline techniques in our experiment.

4.2 Clustering Algorithm

We use an agglomerative clustering algorithm to cluster documents. This hierarchical, clustering algorithm takes a document as input and assigns it to a cluster. Then, all pairwise similarities of all clusters are computed and sorted from largest to smallest. The two clusters having the largest similarity (i.e., most similar) are merged to form a new cluster. The similarities between this new cluster and all other clusters are then computed. This process repeats until one of two conditions is triggered:

1. The number of clusters decreases to a predefine number.
2. The similarity between all clusters is 0, i.e., there is no similarity between any current pairwise clusters. This stopping condition may cause the number of output clusters to be larger than the predefined number.

A third condition prevents the "over-merging" of clusters by estimating a threshold difference, *Est.Diff*, between the sizes of input categories. Using 0.5 as a constant works only when the estimated differences among the sizes of document categories are not large:

$$Est.Diff = \frac{(total\ number\ of\ doc)}{(number\ of\ predefined\ clusters)} * 0.5$$

3. If the sizes of both clusters are larger than *Est. Diff.* value then the agglomerative algorithm will not merge them but will continue with the next pair of clusters.

The following example explains how the stopping conditions work. Assume there are 40 documents and a predefined cluster number of 2. If we only have conditions 1 and 2, then after iteration *n*, the following shows the new clusters formed:

Iteration *n*:
Cluster 1: {'coffee': 1, 'sugar': 1}
Cluster 2: {'coffee': 4, 'sugar': 1}
Cluster 3: {'coffee': 0, 'sugar': 18}
Cluster 4: {'coffee': 15, 'sugar': 0}

Since a cluster change occurred, the *n+1* iteration results are:

Iteration *n+1*:
Cluster 1: {'coffee': 1, 'sugar': 1}
Cluster 2: {'coffee': 4, 'sugar': 1}
Cluster 3: {'coffee': 15, 'sugar': 18}

This iteration "over merges" clusters 4 and 5 due to the influence of outliers. If no changes are made in the next iteration, the accuracy of this clustering is (4+18)/40=55%. Using condition 3 resolves this improper merge. Using the third condition, the calculated estimated difference in sizes is 10 and in iteration *n+1*, condition 3 prevents the merger of clusters 4 and 5 since they both have sizes greater than 10. With no other merges found, condition 2 triggers ends clustering with an accuracy of (15+18)/40=82.5%.

4.3 Datasets

We chose to use the document collection used by [15]. This is from Reuters-21578 Text Categorization Collection in UCI KDD archive [22]. This collection contains newswire documents from 1987. The sizes of the documents range from 12 to 900 words. In [15] size was important. For us, it was not a consideration in document selection. The Reuters-21578 Collection categorizes each of the 21,500 files into 132 categories. A document can contain multiple categories. For instance, a document with two topic tags, "oilseed" and "veg-oil" means it can be classified into either "oilseed" or "veg-oil" category. Document selection randomly collected documents with single and multiple categories. The process avoided any category combinations that did not provide a sufficient set of documents with either single or multiple

categories. In Table 2 we've listed the characteristics of six datasets we used in the study.

Table 2. Document Datasets Used in Study

Dataset Number	Categories used	Number of documents
T-1	gnp, trade	100 docs/category
T-2	dlr, earn, money-supply	50 docs/category
T-6	money-fx, trade	50 docs/category
T-8	wheat, sugar, cpi, trade	80 docs/category
T-10	sugar, coffee	80 docs/category
T-12	ship, interest, gold	50 docs/category

Datasets T-1, T-6, and T-10 identified documents in the Reuters-21578 Collection that contain two categories. Categories selected have some type of association. For instance, the categories 'coffee' and 'sugar' in T-10 included articles that contained both terms when describing how people drink coffee. We also varied the number of documents to see any impact on accuracy or performance.

Datasets T-2 and T-12 used three associated categories with 150 documents evenly distributed in the datasets. We also created one dataset, T8, containing 320 documents that identified four different categories. We collected other datasets but have not completed analysis.

To ensure the distribution of meronyms in each dataset would not influence the accuracy calculation, we used a 5-fold cross-validation process for all cases. Since these are stratified cross-validations where each category had an equal number of documents, we are able to use an average of the runs to compute accuracy [23]. This is the value displayed in Tables 3 through 5.

4.4 Clustering Results Comparison

Comparison without Meronyms

First, we established a baseline similar to other experiments ([5], [3]). Only synonyms and IS-A (hypernym) relationships were used to represent the document. As in previous work, the results in Table 3 show that background knowledge increases accuracy with additional useful information when compared to VSM and TF-IDF methods.

Table 3. Baseline without Meronyms

Dataset Number	VSM	TF-IDF	Concept Chain without Meronyms
T-1	51.5%	69.5%	**76.4%**
T-2	57.3%	78.7%	**83.2%**
T-6	51.0%	51.0%	**73.5%**
T-8	31.3%	51.6%	**69.3%**
T-10	73.0%	75.0%	**88.8%**
T-12	46.7%	56.0%	**69.3%**

Comparison with Meronyms

Next, we examined the impact of including meronyms into our concept chains. If a meronym relationship was discovered, its word frequency was incremented by one (1), carrying the equivalent weight of a synonym. As seen in table 4, all datasets found the inclusion of meronyms to produce more accurate results than VSM or TF-IDF.

Further examination shows half the datasets (T-6, T-10 and T-12) produced poorer accuracy when compared to the concept chain algorithm that excluded meronyms. Table 4 results support previous conclusions that a meronym relationship influences accuracy in a weaker, less reliable manner than synonym and hypernym relationships ([24], [2]). The following two sentences show how a meronym could add noise into the accuracy:

"A car has a window." verses "A car has a windshield."

Both sentences specify a meronym but in the second sentence, the word 'windshield' has a closer relationship to 'car'. This implies it is a more significant noun than the word 'window'.

Additionally the word 'window' is not synonymous with 'windshield' in WordNet. Therefore, we do not capture any meronym. If term frequency of 'window' is high enough to create a concept chain, it never merges with the chain about cars and can potentially add noise to the text characterization, reducing accuracy.

Table 4. Added Meronyms at full weight

Dataset Number	VSM	TF-IDF	Concept Chain with Meronyms
T-1	51.5%	69.5%	**78.8%**
T-2	57.3%	78.7%	**83.3%**
T-6	51.0%	51.0%	**72.0%**
T-8	31.3%	51.6%	**73.3%**
T-10	73.0%	75.0%	**85.8%**
T-12	46.7%	56.0%	**66.2%**

Comparison with weighted meronyms

Most studies consider hypernym and synonym relationships to have stronger or more reliable relationships [2]. With this premise, we decided to explore fractional weights for meronym relationships to see if incremental increases could reduce noise and constraints to document characterization while adding useful information.

We re-ran the same datasets but altered the weight associated with the frequency count of a meronym relationship. The weights used to calculate the contribution of a meronym to the concept chain were 0.25, 0.5, and 0.75. Table 5 shows the results of all weights of meronyms including concept chains with no meronyms (column titled Concept Chain 0) and placing meronyms on equal footing to synonyms and hypernyms (column titled Concept Chain 1).

Table 5. Weighted Meronyms

Dataset	Concept Chain (0)	Concept Chain (0.25)	Concept Chain (0.5)	Concept Chain (0.75)	Concept Chain (1)
T-1	76.4%	72.9%	**82.6%**	79.4%	78.8%
T-2	83.2%	**85.3%**	82.5%	83.3%	83.3%
T-6	73.5%	70.8%	68.3%	**73.8%**	72.0%
T-8	69.3%	70.0%	**74.6%**	71.5%	73.3%
T-10	88.8%	88.6%	90.04%	**91.0%**	85.8%
T-12	69.3%	69.3%	**69.8%**	67.5%	66.2%

The results show meronyms provide additive information to the characterization of documents. All accuracy values show improvement with the meronym contribution if we use a constant less than 1 in calculating the frequency count of a meronym, but

the degree of improvement varies with the weighted value used. We speculate this is due to the different types of meronyms contained in each document and suggest further experiments to analyze the influence of the 3-types of meronyms classified in WordNet. Our assumption is the different types of meronyms provide different levels of useful information for document characterization. The more informative meronyms contribute to improve accuracy but the less informative may add noise. This can over constrain text characterization, making it difficult to identify the major concepts found in a document.

4.5 Runtime Comparison

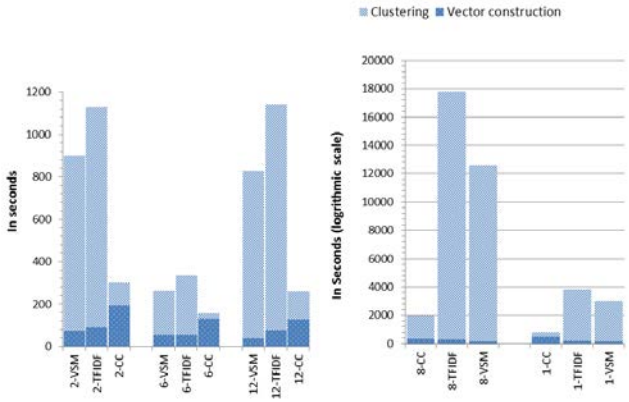
In addition to analyzing accuracy of utilizing meronyms with concepts, we captured runtime performance information shown in Figure 4. From the graphs, we see that combined vector construction and clustering time of VSM and TF-IDF is longer than concept chain (CC) performance. We notice clustering time takes a much longer time in VSM and TF-IDF. This is most likely due to the longer vectors lengths in this representation verses the concept chain sizes due to the purification phase.

In VSM and TF-IDF, a vector will contain many low frequency term counts that need to be analyzed during the clustering process. In the concept chain algorithm, these low frequency terms are pruned out during the purification phase based on the specified threshold as described in section 3.4. Fewer terms in any algorithm will mean a shorter clustering phase. However, the concept chain construction and purification steps in the concept chain algorithm increase vector construction time.

Additionally, TF-IDF and VSM vectors include all non-stopwords which were not stemmed by WordNet's morphology function. This lengthens the vectors of VSM and TF-IDF and thus increasing the cluster time.

It is also easy to see that datasets that contained larger numbers of documents will also have longer clustering times. T-1 has 200 documents (100 for each category) and T-8 has 320 documents Since TF-IDF and VSM will have longer vectors due to the previous reasons stated, the clustering times will also increase as is shown in Figure 4.

Figure 4. Time perfomance analysis



5. CONCLUSION AND FUTURE WORK

Text categorization in concept extraction has employed synonym and hypernym linguistic relationships to improve accuracy in document representation but with the explosion of digital content we need more accurate methods. In this study, we propose a method to capture concepts as a set of chains to represent the significant concepts in a document. This novel algorithm incorporates meronym relationships to provide new information for better text characterization.

The results show meronyms provide additive information to the characterization of documents. We also confirm meronyms provide different levels of useful information for document characterization that is not as strong as synonyms and hypernyms. We speculate the more informative meronyms contribute to improve accuracy but the less informative may add noise. This can over constrain text characterization, making it difficult to identify the major concepts found in a document.

We recognize there are limitations to this study that future work will investigate. A replacement to the current WSD method to provide better context analysis in the decision-making will eliminate noise and provide useful information to the concept chain. We also will investigate WordNet's characterization of meronym types to improve accuracy. Additionally, we will examine other similarity measures to understand their impact on accuracy. Finally, we plan to use the concept chain algorithm in an educational domain to confirm it is not a domain-specific algorithm.

Our intent with this future work is to utilize the concept chain algorithm to characterize digital resources in the support of cataloging efforts. We also see the potential of this algorithm in information retrieval applications to support user needs.

6. ACKNOWLEDGMENTS

Partial support for this research was provided by the National Science Foundation under grant DUE-1043647 and by the New Jersey Institute of Technology.

7. REFERENCES

1. Boast, R., Bravo, M. and Srinivasan, R. Return to Babel: Emergent Diversity, Digital Resources, and Local Knowledge. *The Information Society*, 23, 5 2007), 395-403.
2. Zheng, H.-T., Kang, B.-Y. and Kim, H.-G. Exploiting noun phrases and semantic relationships for text document clustering. *Inf. Sci.*, 179, 13 2009), 2249-2262.
3. Wang, J. Z. and Taylor, W. Concept Forest: A New Ontology-assisted Text Document Similarity Measurement Method. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '07)* (Washington, DC, USA, 2007). IEEE Computer Society.
4. Snow, R., Jurafsky, D. and Ng, A. Y. *Learning syntactic patterns for automatic hypernym discovery*. City, 2004.
5. Hotho, A., Staab, S. and Stumme, G. *Wordnet improves Text Document Clustering*. City, 2003.
6. Elberrichi, Z., Rahmoun, A. and Bentaalah, M. A. Using WordNet for Text Categorization. *The International Arab Journal of Information Technology*, 5, 1 2008), 16-24.
7. Tseng, Y.-H., Lin, C.-J. and Lin, Y.-I. Text mining techniques for patent analysis. *Inf. Process. Manage.*, 43, 5 2007), 1216-1247.
8. Arvidsson, F. and Flycht-Eriksson, A. *Ontologies I*. City, 2002.
9. Girju, R., Badulescu, A. and Moldovan, D. Automatic Discovery of Part-Whole Relations. *Comput. Linguist.*, 32, 1 2006), 83-135.

10. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3, 4 1990), 235-244.
11. Rodriguez, M. D. B., Hidalgo, J. M. G. and Agudo, B. D. Using WordNet to Complement Training Information in Text Categorization. In *Proceedings of the Second International Conference on Recent Advances in Natural Language Processing (RANLP)* (Stanford CA USA, 1997). John Benjamins Publishing.
12. Green, S. J. Building Hypertext Links By Computing Semantic Similarity. *IEEE Trans. on Knowl. and Data Eng.*, 11, 5 1999), 713-730.
13. Hung, C. and Wermter, S. Neural Network Based Document Clustering Using WordNet Ontologies. *Int. J. Hybrid Intell. Syst.*, 1, 3,4 2004), 127-142.
14. Reforgiato Recupero, D. A new unsupervised method for document clustering by using WordNet lexical and conceptual relations. *Information Retrieval*, 10, 6 2007), 563-579.
15. Wang, J. Z. and Taylor, W. Concept Forest: A New Ontology-assisted Text Document Similarity Measurement Method. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence* (2007). IEEE Computer Society.
16. Yang, H. and Callan, J. A metric-based framework for automatic taxonomy induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1* (Suntec, Singapore, 2009). Association for Computational Linguistics.
17. Basu, S., Mooney, R. J., Pasupuleti, K. V. and Ghosh, J. Evaluating the novelty of text-mined rules using lexical knowledge. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (San Francisco, California, 2001). ACM.
18. Berland, M. and Charniak, E. Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (College Park, Maryland, 1999). Association for Computational Linguistics.
19. *Natural Language Toolkit (NLTK)*. Bird, Steven (site is maintained by), City, 2011.
20. Salton, G., Wong, A. and Yang, C. S. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 8, 11 1975), 613-620.
21. Spärck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28, 1 1972), 11-21.
22. Collection, R.-T. C., City.
23. Forman, G. and Scholz, M. *Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement*. City, 2010.
24. Yang, H. and Callan, J. *A metric-based framework for automatic taxonomy induction*. Association for Computational Linguistics, City, 2009.