Biased-Incremental Clustering: A Flexible Knowledge Extraction Algorithm

Kwabena A. Nuamah and Li Fu

***Abstract*—Terminology clustering plays a critical role in the extraction of knowledge from unstructured text data. We present Biased-Incremental Clustering which is designed to make such concept extraction very flexible and human-like. Our approach incrementally clusters concepts by allowing the injection of prior (or existing) concepts into the learner to bias the acquisition of new concepts. It is an unsupervised learning method using Semantic Vectors, Random Indexing and the Word Space (Vector Space) model to perform computations on the concepts. The key aspects of the algorithm run in linear time by using K-Means and a slightly modified version of Bisecting K-Means algorithm. Results show that the Biased-Incremental Clustering algorithm performs well in extracting and clustering terminology from text data containing information that covers both similar and varying domains of knowledge.**

***Index Terms*—Clustering, semantic vectors, machine learning, random indexing**

I. Introduction

An inevitable result of the rapid expansion of the internet is the constant creation of extremely large quantities of data. Consumption of data from news websites, enterprise websites, blogs and social networks among others has become an integral part of our daily lives. With such colossal data sizes available for processing, it has become more difficult to find specific information we want from the large amounts of free-form (unstructured/semi-structured) text data.

This paper introduces the Biased-Incremental Clustering method to terminology (concept) using an incremental and hierarchical approach. It is influenced by the natural information discovery process that humans exhibit, allowing it to bias its learning experience with prior knowledge it might possess. Such a learner can therefore start off as a "novice" in a particular domain using an exploratory approach to understand the relationships and similarities (or dissimilarities) between concepts. This helps the learner cover a wider surface area of knowledge from multiple domains. The clustering system also has the flexibility to alter the *prior knowledge bias* such that it places more emphasis on some specific knowledge it has about a domain of knowledge. Learning and clustering under such a bias often leads it to gain deeper information about a specific domain from newer datasets it might encounter.

This clustering approach makes use of Distributional Semantics and Vector Space Models [1] to make mathematical representations and computations of the content of documents relatively easier from a theoretical point of view. To ensure that the clustering time scales well with large quantities of data, the algorithm used variants of the K-Means and Bisecting K-Means algorithms with linear and near-linear time complexities respectively.

Manuscript received February 22, 2012; revised April 24, 2012.

The authors are with the School of Software Engineering, Chongqing University, Chongqing, China (e-mail: kwabena.nuamah@gmail.com, tel.: +8613883491754).

II. Semantics

A growing need to make computers understand free-form text has led to greater focus on the concept of meaning or semantics. We are not only simply interested in the meaning of text and information from the human point of view, but need to find formal mathematical representations and the corresponding mathematical operations. The goal is to make computer systems obtain meaning from the vast amounts of free-form textual information available from sources such as the Internet. Our ability to mathematically compute semantics is an important step in automating many of the recent "semantic" objectives on the internet. For instance, the objective of creating the Semantic Web [2] depends on the extent to which ontologies of domains can be created automatically in an unsupervised way.

The Vector Space Models [1] and Semantic Vectors Package [3] based on Random Indexing [1] (Random Projection [4]) provide a means of mathematically representing the content of a free-form text. They create vectors (or points) in a mathematical space in an entirely unsupervised process, and this representation is learned from text in such a way that concepts with similar or related meanings are near to one another in that space. They also achieve this representation in a computationally tractable manner since they apply Random Indexing which deals with dimension reduction. Biased-Incremental Clustering is built on this model.

III. Biased-Incremental Clustering

Biased-Incremental Clustering, introduced in this paper, is an unsupervised learning algorithm designed to improve the semantic learning of information from textual documents. It is based on the premise that, information retrieval systems accumulate some information about the domain which can be used to improve the subsequent learning experience. The 'prior' acquired (or existing) knowledge will be semantically used to filter out relevant knowledge from the corpus being consumed. The extent to which prior knowledge influences the learning experience is determined by a bias (λ). The key objective of the Biased-Incremental Clustering algorithm is

to semantically cluster the terms or concepts in the data.

Biased-Incremental Clustering is inspired by the natural process of learning by humans. We often learn by making use of our prior knowledge and also learn to enhance a specific domain of knowledge by filtering out and assimilating only the relevant bits of information. Again, our learning experience is based not solely on the syntax of the content of the documents, but more on the semantics of the corpus.

A. Implementation of Biased-Incremental Clustering

In order to achieve linear or near-linear complexities in execution, the critical components of the clustering algorithm is a combination of the K-Means and Bisecting K-Means algorithms. Prior to the clustering stage, the indexing phase is accomplished with fast modules which also comfortably take care of the high dimensions of unstructured textual data. Apache Lucene [5] and Semantic Vector Package [6] are used for this. The Biased-Incremental Clustering algorithm is in two main steps (as shown in Fig. 1): the Indexing step and the Clustering step.

1) Indexing Step: This begins with the use of Apache Lucene to index the text. It builds the standard term-document index. Apache Lucene also facilitates tokenization of terms in text. It subsequently generates semantic vectors using the Semantic Vectors Package (SVP). It generates a semantic representation of the terms in a mathematical space such that the relationships between terms and concepts can be mathematically computed. The SVP helps realize this goal by building the distributional semantic models by applying Random Projection based on the work of Sahlgren [1] and Kanerva et. al [7].

2) Clustering Step: The clustering stage of the Biased-Incremental Clustering begins by running the K-Means algorithm on the semantic vectors which were created by the Semantic Vectors Package. The algorithm then selects a total of T terms that are closest to the centroids in each cluster. This is followed by identifying the λ-Nearest Terms in the semantic vector space, choosing the set of terms from the 'prior' knowledge as centroids. These terms will be used to bias the clustering process. The T terms and the λ bias terms are merged into a sub-vector space. Finally, the newly created semantic sub-vector space is clustered hierarchically using a variant of the Bisecting K-Means algorithm introduced by Steinbach, et al. in [8].

B. Clustering in the Biased-Incremental Clustering Algorithm

The clustering step involves the use of K-Means, Nearest-Terms and Bisecting K-Means algorithm. First, a flat (partitional) cluster is created using the semantic vector of terms that were generated by the SVP in the indexing step. The K-Means algorithm is employed for the initial clustering of terms. This initial clustering finds terms that are semantically similar or closely related, then groups them into clusters.

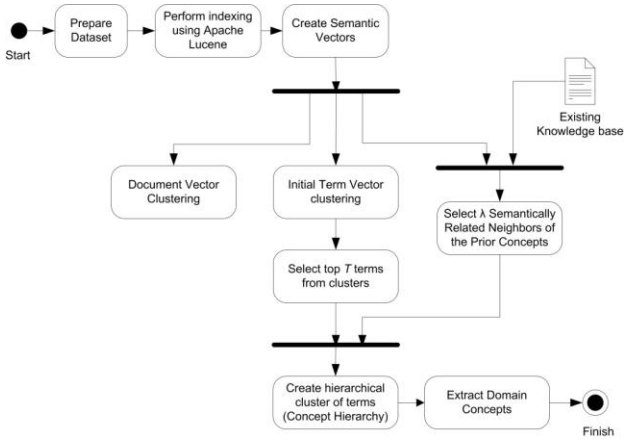

Fig. 1. Biased-incremental clustering process flow.

The K-Means algorithm runs a number of trials, with each trial randomly choosing vectors as the initial centroids of the cluster. A function assigns term vectors to the closest centroid vector that it is most similar to. The result is a group of clusters containing semantically related terms. In order to accommodate the random initialization of the K-Means algorithm, a parameter is passed to the function to determine number of times (trials) the algorithm should run, with each run initialized to randomly selected centroids. The 'best' is selected from the set of trial clustering results. The best clustering is determined by finding the trial clustering having the minimum total Euclidean distance of each term from its cluster's centroid.

C. An Improved Measure of Similarity

The measure used to determine the cluster association of a term vector is a function based on the amount of '*Semantic Force*' between vectors and the Cosine Similarity measure between two vectors. The Semantic Force between two term vectors (in Fig.2) is derived from Newton's Universal Law of Gravitation which states that the force of attraction between any two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. The concept of Semantic Force is motivated by the works of Philosopher Ludwig Wittgenstein and Linguist John R. Firth.

Wittgenstein's investigations in [9] shows that the meaning of a word is correlated with its use in the language and is the object for which the word stands. Firth's theory [10] is the context dependent nature of meaning which he expresses as: "*You shall know a word by the company it keeps.*" Semantic Force therefore considers each term in a corpus as an object exerting a force on other terms in the corpus. Two terms with high semantic similarity or relatedness will have a stronger Semantic Force between them. Conversely, two non-related terms in a corpus will have a weak Semantic Force between them.

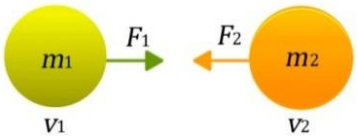

Fig. 2. Semantic force between terms.

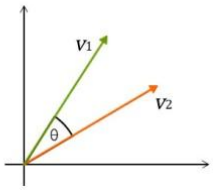

Fig. 3. Cosine similarity using semantic vectors.

The Semantic Force, *F*, between any two term vectors is calculated as:

$$F_1 = F_2 = F(v_1, v_2) = \frac{m_1 m_2}{|r|^2} \quad \text{with} \quad m_i = \log\frac{N}{f_i}$$

where m_1 and m_2 are the respective masses of vectors v_1 and v_2 , r is the Euclidean distance between vectors v_1 and v_2 , N is the total number of terms, and f_i is the global frequency of term *i*.

Cosine Similarity is the measure of the angle between two vectors as shown in Fig 3. The Cosine Similarity Measure (i.e. cos(θ)), *C*, between two term vectors v_1 and v_2 is calculated as:

$$C(v_1, v_2) = \frac{v_1 \bullet v_2}{\|v_1\|\|v_2\|}$$

where $v_1 \bullet v_2$ is the dot product of the two vectors v_1 and v_2 , and $\|v_i\|$ *is the Euclidean Norm*(or length) of the vector v_i .

Therefore the measure of association of a vector to a centroid (its cluster mapping) is determined by the function

$$S(v_1, v_2) = F(v_1, v_2) \,.\, C(v_1, v_2)$$

$$S(v_1, v_2) = \frac{m_1 m_2}{|r|^2} \cdot \frac{v_1 \bullet v_2}{\|v_1\|\|v_2\|} = \frac{m_1 m_2}{|v_2 - v_1|^2} \cdot \frac{v_1 \bullet v_2}{\|v_1\|\|v_2\|} = \frac{(\log\frac{N}{f_1})(\log\frac{N}{f_2})}{|v_2 - v_1|^2} \cdot \frac{v_1 \bullet v_2}{\|v_1\|\|v_2\|}$$

The Cosine Similarity measure alone is inadequate when searching for key terms or topics in a free-form corpus. The Cosine similarity measure is based on the semantic relatedness of the vectors, but not very useful when the significance of a term within the context of the corpus is of importance. By incorporating the semantic force of attraction between the terms vectors (or a centroid and the terms around it), the mapping of a vector to a cluster is based not only on semantic similarity, but also on the semantic importance or relevance of the terms. This is primarily due to the fact that the measure of the force of attraction is based on the mass of the vector, which is a function of the importance of a term in the corpus.

The next step identifies a maximum of *T* terms that are closest to the centroids. The function takes a parameter *t* = *T*/(number of centroids). The function uses the improved similarity measure above to determine the nearest terms to the centroid terms. A priority queue is used to make it much faster to identify the terms having the largest similarity with the centroid term.

Next is the selection of terms close to the bias terms. The bias terms are obtained from the prior knowledge that exists. These bias terms will be represented as an array of string terms and serve as centroid vectors. The use of the bias terms skews the extraction and clustering operations in favor of the prior knowledge. The algorithm will attempt to find λ most semantically related terms to the bias terms.

The value of *T* and λ chosen further reduces the volume of the data to be clustered in the final, hierarchical clustering step. The size of the data *S* available for the hierarchical clustering is: $S = T + \lambda$.

Now that the most representative terms of the initial flat clustering's resultant clusters and the terms most similar or related to the bias terms have been merged, the algorithm finally re-clusters the merged terms hierarchically. The resulting structure represents the hierarchical structure of the concepts. To achieve this clustering operation in a fast, linear or near-linear time, a variant of the Bisecting K-Means algorithm, introduced in [8], is used.

The result of the hierarchical clustering using the Bisecting K-Means algorithm is a binary tree. Each bisection creates two clusters which are added as child nodes in the cluster tree. The smaller-sized cluster is added as the right child of the node that was split. The larger one is added as the left child. Each node of the binary tree will represent a cluster of semantically-related terms. The leaves of the binary tree will therefore represent the result of the entire hierarchical clustering process. If required number of clusters is *k*, then the array representation of the binary tree must have a capacity of *2k-1*. What makes the Bisecting K-Means fast is that each iteration deals with a smaller set of term vectors to cluster, and so it begins to speed up as it progress. A parameter, *numTrials*, determines how many times the 'internal bisection' should occur, after which the trial 'bisection' with the best intra-cluster similarity (similar to that of the K-Means previously discussed) is selected. Other parameters determine how the algorithm behaves during execution.

IV. Evaluation and Analysis

A. The Bias Function

The bias (λ) function determines the level of influence that the prior knowledge has on the present clustering operation. The different values of the λ-function can be used to interpret the nature of the machine learning experience. Generally, there are four scenarios for the value that λ can take:

$\lambda = 0$: There is no prior knowledge with which to influence the present learning experience.

$\lambda < T$: There is very little prior knowledge which could be used to influence the knowledge acquisition. The learner is therefore focused on finding new concepts from the corpus.

$\lambda = T$: The influence of prior knowledge is relatively

proportional to the new knowledge discovered. This is a balance between the search for new knowledge and the search for knowledge similar to what is already known.

$\lambda > T$: There is more emphasis on what is already known. The goal is therefore to refine the existing knowledge by adding only the most relevant concepts similar to the existing knowledge.

B. Evaluation

TABLE I: CLUSTERS GENERATED FROM SCI-MED DATA SET WITH NO BIAS TERMS

Cluster 1	Cluster 2	Cluster 3	Cluster 4
mother	louisiana	antibiotic	salvador
ultrasound	centrafricaine	antibiotics	chile
	cameroun	trials	honduras
	research	immunity	trrinidad
	benin	computational	bermudes
	scientific	output	antigua
	medical		islands
	chad		
	article		
	news		
	newsletter		

TABLE II: CLUSTERS GENERATED FROM SCI-MED DATA SET WITH BIAS TERMS *'VACCINATION, VACCINE, VIRUS'*

Cluster 1	Cluster 2	Cluster 3	Cluster 4
antibody	oral	virology	emergence
impurities	parasites	influenzae	agglutination
	meningitis	immunologic	aggregation
	acellular	hiv	membrane
	vaccination	toxoids	cells
	vaccines	immunodeficiency	mucus
	vaccine	immunization	moisture
	infants	virus	
	doses	cytotoxic	
		antigen	

TABLE III: CLUSTERS GENERATED FROM SCI-MED DATA SET USING TERMS FROM CLUSTER 2 OF TABLE 2 AS BIAS TERMS

Cluster 1	Cluster 2	Cluster 3	Cluster 4
meningococcal	antigen	immunodeficiency	immunology
miningococcus	intravenous	supplementary	virology
meningitids	dose	vaccine	oral
cytotoxic	infants	manufacture	tetanus
epidemics	illnesses	manufacturer	immunologic
agglutination	diarrhea	hiv	immunization
virus	diphtheria	acellular	meningitis
vaccination	proneness		children
biology	crossreactive		doses
vaccines	influenza		
	booster		

Tables 1, 2 and 3 show how the Biased-Incremental Clustering approach is used to build and refine a knowledge-base. The *Sci-Med* sub-dataset of the 20 Newsgroups Datasets [11] was used. Table 1 shows how relevant terms are still identified and clustered together, even without bias terms. Table 2 introduces some bias terms, and the resulting concepts in cluster 2 of Table 2 are used as bias terms to drill deeper in the data to find more knowledge about concepts in cluster 2. It is important to note that the experiments conducted extracted the relevant concepts from a very small sample of the entire dataset. 15,094 semantic vector terms were created for the *Sci-Med* sub-data set. The Biased-Incremental Algorithm obtained the clusters in Tables 1, 2 and 3 from just 200 semantic vectors.

The F-Measure was also calculated using a dataset made up of some books from Project Gutenberg[1]. We used 10 full texts from the medical, agriculture and technology bookshelves. Twelve bias terms (4 for each text domain) were used to run Biased-Incremental Clustering. The F-Measure was determined first for each cluster using the corpus domain that best represented the cluster.

TABLE I: F-MEASURE EVALUATION

T=Total terms extracted in cluster
P=Precision, R=Recall
F_1=F-Measure with equal weights on P and R
$F_{0.5}$=F-Measure with P twice as important as R

Clusters (dominant domain terms)	T	P	R	F_1	$F_{0.5}$
Cluster 1 (23 TECH terms)	25	0.92	0.70	0.79	0.86
Cluster 2 (27 MED terms)	27	1.00	0.52	0.68	0.84
Cluster 3 (23 MED terms)	23	1.00	0.44	0.61	0.80
Cluster 4 (34 AGRIC terms)	35	0.94	0.97	0.96	0.95
Averages		**0.97**	**0.66**	**0.76**	**0.86**

C. Synonyms Versus Semantic-Relatedness

In the Biased-Incremental Clustering algorithm, emphasis is on identifying terms with significant similarity or relatedness within the context of the corpus. For instance, in Table 2, the term 'infant' and 'dose' are in a different cluster from that of terms 'children' and 'doses'. In the English language, the words 'children' and 'infants' are synonyms, as are the terms 'dose' and 'doses'. However, within the *Sci-Med* dataset, 'children' and 'infants' are used in different contexts, resulting in the observed clusters.

V. CONCLUSION

The Biased-Incremental Clustering algorithm presented here makes the extraction of terminology from text datasets containing related or unrelated domains of knowledge very flexible and natural. The algorithm is capable of using new concepts it learns as well as existing knowledge to "bias" its future learning tasks. The improved measure of similarity between concepts yields very good precision in clusters. It performs well in extracting and clustering relevant and related concepts in an unsupervised manner, making it desirable as a major component for tasks such as ontology building.

ACKNOWLEDGEMENT

This paper is dedicated to the Late Sampson K.A. Nuamah for his immense support during the research and preparation of this paper.

REFERENCES

[1] M. Sahlgren, "An Introduction to Random Indexing," 2005.
[2] T. B. Lee. J. Hendler, and O. Lassila. (May 2001) Scientific American. [Online]. Available: http://www.scientificamerican.com/article.cfm?id=the-semantic-web

[1] Corpus from Project Gutenberg obtained from http://www.gutenberg.org

[3] D. Widdows and K. Ferraro, "Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application,".
[4] P. Kanerva, *Sparse Distributed Memory*, MIT Press, 1988.
[5] The Apache Software Foundation. (2010) Apache Lucene. [Online]. Available: http://lucene.apache.org/java/docs/index.html
[6] D. Widdows and T. Cohen, "The Semantic Vectors Package: New Algorithms and Public Tools For Distributional Semantics," 2010.
[7] P. Kanerva, J. Kristofersson, and A. Holst, "Random indexing of text samples for latent semantic analysis," in *22nd Annual Conference of the Cognitive Science Society*, 2000.
[8] M. Steinbach, G. Karypis, and V. Kumar, "A Comparison of Document Clustering Techniques,".
[9] Wikipedia. (2011) Wikipedia. [Online]. Avaiable: http://en.wikipedia.org/wiki/Philosophical_Investigations
[10] Wikipedia. (2011) Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/John_Rupert_Firth
[11] MIT CSAIL. (2011) Homepage for 20 Newsgroups Data Set. [Online]. Available: http://people.csail.mit.edu/jrennie/20Newsgroups/