Preprint: In proceedings of 6th IEEE International Conference on Big Data Security on Cloud (BigDataSecurity 2020)

Affinity Propagation Initialisation Based Proximity Clustering For Labeling in Natural Language Based Big Data Systems

Adithya Bandi
Department of Computer Science
University of Maryland, Baltimore County
Baltimore, MD, USA
adithyb1@umbc.edu

Karuna Joshi
Department of Information Systems
University of Maryland, Baltimore County
Baltimore, MD, USA
kjoshi1@umbc.edu

Varish Mulwad
Artificial Intelligence
GE Research
Niskayuna, NY, USA
varish.mulwad@ge.com

***Abstract*—A key challenge for natural language based large text data is automatically extracting knowledge, in terms of entities and relations, embedded in it. State of the art relation extraction systems requires large amounts of labeled data, which is costly and very difficult, especially in industrial settings, due to time constraints of subject matter experts. Techniques like distant supervision require the availability of a related knowledge base, which is rarely possible. We have developed a novel model for automatically clustering textual Big Data, based on techniques inspired from Active Learning and Clustering, that can derive powerful insights and make the data ready for machine learning with minimal manual effort. Our approach differs from Active Learning as we operate under weak supervision, where all the instances provided for training are not manually labeled. Secondly, This differs from any prevailing clustering algorithms as we adopt a whole new approach of proximity clustering based on affinity propagation. Due to the extrapolation of the labeling efforts, our model makes it easier to adopt deep learning approaches with minimal manual effort. In this paper, we describe our algorithm in detail, along with the experimental results obtained for them.**

***Index Terms*—relation extraction, unsupervised labeling, clustering, affinity propagation, Labeling, Natural Language Processing, Big Data**

I. Introduction

Text-based natural language format constitutes a considerable portion of Big Data stored in various systems, so it has become imperative to have systems process it as efficiently as they process numerical or categorical data. Natural Language Processing (NLP) algorithms for tasks such as named entity recognition and linking, relation classification and extraction, text classification, and the like have shown massive improvements in the past few years. Most of these improvements have come in form of approaches based on supervised deep-learning algorithms. A key ingredient for the success of these algorithms has been the availability of large scale labeled datasets often comprising tens of thousands of labels. With the ease of availability of these NLP algorithms in the form of open-source software packages, researchers have begun exploring its applicability in a variety of domains, from cybersecurity to industrial to healthcare. While a large quantity of text data is available in these domains in the form of maintenance records, medical literature, and security blogs, the challenge remains in labeling them for state-of-the-art NLP algorithms to work well.

Manually labeling the data is a burdensome process. Subject matter experts can label only limited amount of data and will not be able to match the scale required for the current NLP algorithms to work efficiently. A possible solution could be the use of Amazon Mechanical Turk, a crowd-sourcing website that allows users to perform a variety of tasks. Researchers have successfully used it to label large quantities of data. However, this may not be feasible for several domains due to data privacy and security issues, besides the lack of domain expertise amongst the users performing the task on the Amazon Mechanical Turk platform.

Automated dataset labeling has been the focus of ongoing research initiatives, leading to the development of various paradigms to tackle the problem. Distant supervision [8] leverages existing knowledge bases (such as Freebase) to label datasets automatically. This approach has shown success for labeling datasets used in developing supervised models for relation extraction. However in the context of domains such as legal and cybersecurity, such pre-existing knowledge bases are not readily available. The Weak supervision [1] paradigm, on the other hand, relies on programmers or researchers to write labeling, data augmentation, and data transformation functions to acquire weak labels. This approach heavily relies on the programmer's ability to write proper functions providing broad coverage over the dataset.

In this paper, we present a novel approach that overcomes the need for existing knowledge bases as well as the need for programmers to develop custom labeling functions. We draw our inspiration from active learning and unsupervised approaches, namely clustering, to automatically label instances in a dataset with minimal user input. Our key contribution is a hierarchical multi-level clustering algorithm that first divides the data into different clusters based on semantic knowledge. Later, each cluster is subject to Affinity Propagation based on the semantic vector representation of corresponding instances, followed by the reduction in the number of clusters with a novel approach to reassign instances to different clusters based on the similarity between them and potential cluster centroids. Finally, we sample clusters to generate a diverse set of samples, which comprises instances ranging from the most dissimilar to most similar with the cluster centroid. The Experts label these instances. A frequency-based approach then determines the appropriate label for each cluster. While a variety of text datasets can be labeled, we use a relation extraction as an exemplar task to demonstrate and evaluate our approach.

Section II of this paper describes the background and related work. In Section III, we describe our approach and algorithm in detail. In Section IV we describe the experimental results and conclude in Section V along with a brief description of our ongoing work.

II. Related Work and Background

We have reviewed existing semi-supervised approaches, distant supervision based approaches, and unsupervised approaches with emphasis on the clustering-based approaches. We have also reviewed the various embedding models that we considered for our approach and list the key approaches below.

A. Semi Supervised Approaches

Brin and Sergey [2] proposed an algorithm named Dual Iterative Pattern Relation Expansion (DIPRE) based on the idea that given a good set of patterns, good set of tuples can be found and vice versa. Basically, it forms regular expressions based on a number of sentences containing particular entity pair,where the regular expressions only model the prefix,suffix and middle and take presence of any terms in the entity position to be a new sample with same relation. Agichtein et al. [3] proposed an algorithm called Snowball similar to DIPRE where the prefix suffix and middle are converted to vector space and similarity measures were used to match different tuples.A pattern is taken as the centroid/mean of the tuples of certain pattern and was assigned a confidence score based on positive and negative matches for a particular pattern.

B. Distant Supervision

The distant supervision approach [8] is generally used to assign labels for unlabelled data. It takes advantage of the available knowledge bases for instance, freebase [9] and DBpedia [10] to assign relations. The relation instances present in the database are considered as facts and any sentence having the same entities is assigned the given relationship in the database.The basic assumption is that the relation between any two entities present in knowledge base would be the same if they appear in any sentence. The distant supervision first proposed by Mintz et al. in in [8] was used for relation extraction with the freebase as a knowledge base for labeling purpose and the wikipedia dataset was used for relation instances.

C. Clustering

One of the main components of our approach is the Clustering. So we have reviewed several clustering algorithms to determine the best approach. Each Clustering algorithm makes assumptions about the underlying data distributions as there is no single criteria for an optimal clustering algorithm in general. These assumptions generally dictate how the similarity of the data points is calculated and form the basis for classification of algorithms.

- Density-Based methods: the basic idea behind these methods is that dense regions can be considered as clusters as the data points with in a region will have more similarity to each other than data points in different regions. These allow for arbitrary-shaped distributions as long as dense areas can be connected. We do derive inspiration for the proximity clustering used in the algorithm from this clustering paradigm.
- Centroid based Clustering: are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. These models run iteratively to find the local optima. These clustering paradigm also serves as an inspiration for the proposed algorithm where the proximity is calculated based on the closeness to a perceived center of the cluster.

Our approach is based on Affinity Propagation. One of the most popular algorithm, similar to Affinity Propagation is K-Means but the main drawbacks of the algorithm are the need to provide the number of clusters as the initial parameter to the algorithm and initialization of the cluster centers. Affinity propagation addresses these drawbacks by taking as input parameters, measure of similarity between pairs of data points and considering all the data points as potential exemplars (cluster centers). The basic flow of the algorithm is message passing by exchanging messages between data points until a high quality of exemplars and their corresponding clusters are emerged.

Clustering Based Approaches: One of the earliest works using clustering approach was proposed in [4], where the named entity pairs are clustered based on the context using hierarchical clustering with complete linkage. The entities are paired based on number of words between them. These are called co-occurring entities and are paired with each other.An unsupervised feature selection for relation extraction was proposed in [5], where they have developed a feature selection method with the task of finding the most important words in a context and removing the uninformative noisy words from the similarity computation. To enable automatic labeling they find the typical and discriminative words across different clusters using a discriminative category matching. Later works took information from web especially by mining wikipedia texts such as in [6] where they form concept pairs by using Wikipedia structure, where current title becomes a principal concept and its paired with secondary concepts obtained from the articles it's been linked to. They proposed a two-step clustering approach from grouping the concept pairs with the same relation type. The concept pairs are clustered using dependency patterns by parsing sentences in Wikipedia articles using a linguistic parser, and surface patterns from redundancy information from the Web corpus by using a search engine. Generative probabilistic models, similar to Latent Dirchlet Allocation based topic models were used in [7] where the relation types play a role similar to topics. The relation instances features were based on the dependency path between them. The model also incorporates the constraints relating to the relation type to the entity types.

D. Embeddings

When it comes to the representation of the text, the naive straight forward approach would be to use the one-hot encodings, where each word would be represented as a dimension. But this is a very sub optimal approach and would face issues due to the vocabulary size and the computational issue. To tackle the above issues, the word embeddings were introduced. It is a word representation where it allows words with similar meaning to be similar to each other in the representation.The similarity between representations is generally evaluated by considering the cosine similarity between them. This has been one of the key breakthroughs in the Natural Language Processing domain and has now become a key component in various applications.

Static word embeddings are the word representations obtained based on training corpus without considering the current context of the word whereas contextual word embeddings, as name suggests does take the current context into consideration during embedding generation .Few examples of Static word embeddings are Word2Vec [12], Glove [13] , Fast Text [14], and lda2Vec [15]. Few examples of contextualized word embeddings include ELMO [16], and Generative Pre-Training2 [17]

1) Glove and Flair Embeddings: The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the Word2Vec [12] method for efficiently learning word vectors, developed in Stanford [13]. Word representations were developed using matrix factorization techniques such as Latent Semantic Analysis (LSA) which perform well using global text statistics but do not succeed at capturing meaning and demonstrating it on tasks like calculating analogies, where algorithms like Word2Vec perform relatively better.GloVe is an approach to combine the global statistics of matrix factorization techniques with the local context-based learning in word2vec.Rather than using a window to define local context, GloVe constructs an explicit word-context or word co-occurrence matrix using statistics across the whole text corpus.

Flair embeddings [21] are contextual string embeddings, which capture the latent syntactic-semantic information. The main differences being that it is trained without any explicit notion of words and thus fundamentally models words as sequences of characters and are contextualized by their surrounding text, meaning that the same word will have different embeddings depending on its contextual use.

In the current work, we use Stacked embeddings, where we combine traditional embeddings with contextual string embeddings. In particular, we combine the Glove embeddings [13] with backward and forward flair embeddings as this is a combination generally recommended [21].

2) Infersent Embeddings: The basic idea in infersent [19] is to use SNLI (Standford Natural Language Inference) data [18] to train a model for Natural Language Inference (NLI) problem. The Stanford Natural Language Inference (SNLI) corpus [18] is a classification task containing 570k pairs of sentences that are written and labelled by humans. Natural Language Inference (NLI) problem is defined as to identify the relationship between two sentences, a hypothesis and a premise. There are three categories which are entailment, contradiction, and neutral.The Infersent primarily consists of encoders, which are trained by extracting the basic features from their output on each pair of sentences and feeding them into a fully connected layers. The sentences are encoded separately to make sure that the encoding would apply to generic sentences, independent of context.

The InferSent encoder runs a forward and backward Long Short Term Memory on the GloVe vectors [13] , concatenates the hidden states for each word, and then applies max-pooling.A sentence is thus mapped to a 4096-dimensional vector.

E. Performance Metrics

Homogeneity, completeness, and V-measure are three key related indicators of the quality of a clustering operation and thus the same have been used to assess the quality of labeling in the current work.The homogeneity conveys whether each cluster contains the members of a single class and completeness conveys whether all members of a given class are assigned to the same cluster. V measure is a harmonic mean of homogeneity and clustering providing more comprehensive view.

III. Technical Approach

Problem Formulation

We use relation extraction as an exemplar task to describe our proposed approach. The objective of this task is to extract semantic relationships that exist between named entities in text. For example, given the sentence “Barack Obama is the 44th president of the United States of America”, the relation extraction task would extract/identify “presidentOf” relation between “Barack Obama” and “United States of America”. For this paper, we assume that named entities in a given sentence are already extracted with the help of existing named entity recognition [21] & linking systems [24]. Given many instances of the form of pair of named entities (e.g. Obama, USA) and a sentence in which they appear (see example sentence above), our objective is to label these instances with relations (e.g. presidentOf). This labeled dataset then can be used to train relation extraction systems with state-of-the-art deep learning approaches [23].

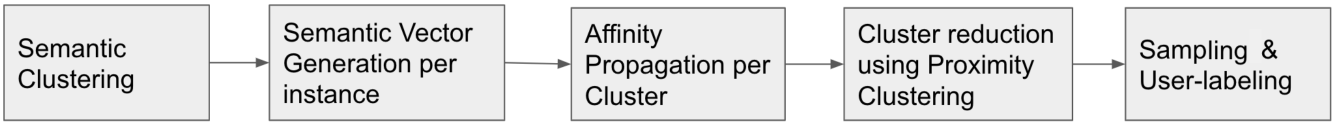
Semantic Clustering
Semantic Vector Generation per instance
Affinity Propagation per Cluster
Cluster reduction using Proximity Clustering
Sampling & User-labeling

Fig. 1. Overview of Our Approach

Approach Overview

Fig 1 gives a high-level overview of our approach. The high-level semantic type information associated with the named entities (e.g. Obama's high-level semantic type is Person) is used to first cluster the instances into different groups. The instances (named entity pairs and the associated sentence) within these clusters are then mapped into a high-dimensional vector space, before applying Affinity Propagation over these vectors to create the next-level of grouping. The number of clusters is further reduced by reassigning instances to different clusters based on their similarity with potential cluster centroids. Finally, a sample of instances from each cluster is chosen and presented to subject matter experts for labeling. In the rest of the section, we describe each component's details.

A. Semantic Clustering

Named entity recognition (NER) systems often produce high-level semantic type information for each extracted entity. Examples of semantic types produced by NER systems include Person, Place, Organization, Software, Means of an Attack, Consequence of an Attack, etc. We leverage type information to form our first-level of clusters. We loop through each entity pair to identify all possible domain and range pairs. For a given relation (e.g. presidentOf), the domain describes the semantic type for the subject (e.g. Person) of the relationship and the range describes the value or object type (e.g. Place). We form the entity pairs for each relation instance by concatenating the entity types of the corresponding entities in the order that they appear in the sentence.Every domain-range pair (e.g. Person - Location) is used to represent a unique cluster. Entity pairs based on their semantic types are assigned to the appropriate clusters. For example, instances (sentences) associated with the entity pair Obama - USA will be assigned to the Person - Location cluster.

B. Text to Vector Mapping

we generate the mathematical vector representation for the context using the embedding models discussed earlier.We generate the embeddings based on the context, that is the words between the entities including themselves. In particular, we use stacked flair based embeddings [21] and glove embeddings [13],where we combine the Glove embeddings [13] with backward and forward flair embeddings. We also generate infersent [19] based embeddings with a goal to compare the performance of approach across various embedding models.

C. Affinity Propagation and Cluster size reduction using Proximity Clustering

We use the Affinity Propagation to further cluster each of the groups obtained in the above step, that is the entity pair clusters .The clusters obtained using Affinity Propagation are used as the initial clusters for the next step of algorithm, more specifically this step provides potential cluster centers along with their priorities as the size of the corresponding cluster.The algorithm can be seen as ensemble of Centroid based clustering and Density based clustering . One of the advantages of Affinity Propagation was more homogeneous clusters were obtained but the number of clusters seemed to be relatively high. The purpose of proximity clustering is to reduce the large number of clusters obtained using Affinity Propagation while increasing the v measure. The cluster centers obtained using Affinity Propagation serve as potential cluster centers. The idea of the proximity clustering is to assign the instances of a cluster to any other cluster, where the distance of the instance to the center of the chosen cluster is less than a predefined threshold. In the current context, the cosine similarity is used with the understanding that the angle between any two vectors as the actual distance between them. For every cluster starting with highest preference, we check for all the relation instances to see if any of them cross a particular threshold and if they do they are assigned to the corresponding cluster. The order of evaluation of clusters is determined by the size of initial clusters where the cluster with highest size is given more preference. The relation instance is probably not assigned to the most optimal cluster, the one with shortest distance from cluster center to the relation instance which differentiates the current approach from any other standard clustering algorithms. The relation instance will be assigned to the last cluster that is encountered during evaluation of the clusters, where distance to the center of the cluster is less than the threshold value.

D. Sampling and User-labeling

Labels are assigned to instances by sampling them from each cluster. We select the most similar and dissimilar instance to the cluster centroid. The reason being that the labeling captures most diverse samples and thus most representative of the extremes of the cluster. The number of similar and dissimilar instances to be sampled is a tunable parameter. Subject matter experts are asked to provide a relation label to each sampled instance. A majority vote is used to determine the appropriate label for each cluster. If the majority vote score is above a certain threshold, the label is propagated to all instances in the cluster, else the cluster (and its instances) are discarded from the labeling process.

IV. Experiments and Results

A. Experimental Setup

The google relation extraction corpus [11] consists of four main parts of information for each relation instance. They are entities information, relation, the evidences and the judgements. The relations we considered for our experiments are place of birth, place of death, education degree and institution. We extract the sentence from the evidences. The subject and object information is stored as Freebase MID's instead of the actual entity names. The mappings from these identifiers to Wikipedia database entities is presented in a separate file. These mapping are first obtained and later we query the corresponding Wikipedia database entity URL to retrieve more information about the entity. The obtained response from the URL is used to retrieve the actual name of the entity. Now regarding the relation, the probable ground truth about the relation present can be obtained by going through judgements information present in each relation instance. As name suggests, the judgements information mainly consists of the judgement provided by each rater, the values being ”yes”, ”no”, and ”skip”. For every relation instance, the majority judgement among the raters is assessed and for a ”yes”, the relation instance is labeled as positive instance and for a ”no”, the relation instance is labeled as negative instance, in a ”skip” case the relation instance is ignored.

At this stage we obtain the subject,object, sentence and labels for each relation instance. For each relation instance, we have also went through the data to assign the entity types for subject and object. The entity pair information is obtained by concatenating the entity types in the order that they appear in the sentence. For extracting the embedding feature, the context is extracted as the words present between entities including them. For certain entities, the entity name retrieved from the corresponding wikipedia database is not present in the sentence so therefore the relation instances are dropped. The total number of relation instances extracted are 16,400.

For each context, the embedding representations are generated. The relation instances are first grouped based on the entity pair and then each of them are clustered based on the embedding presentations using the affinity initialization based proximity clustering algorithm.

B. Performance Evaluation and Insights

For performance evaluation, We consider the homogeneity score, completeness score, v measure, and the number of clusters. The reason being homogeneity score lets us know a measure of data points of each cluster belonging to the same class label, the number of clusters lets us know an estimate of the labeling efforts required by the subject matter experts, the completeness score signifies whether all data-points belonging to the same class are clustered into the same cluster. The v measure score is a harmonic mean of homogeneity and completeness and provides a more comprehensive comparison. As the core approach forms the novel part of the proposed framework, the baseline considered is the proposed framework with Affinity Propagation without the core approach. Affinity Propagation is one of the prominent algorithms, used when the number of underlying clusters is unknown.

One of the hyperparameters present in the algorithm is the threshold in terms of the distance between any two embedding vector representations, which informs the proximity. Now we have a unique problem in evaluating the hyperparameter as we do not have any labels hence no validation set. So we propose a reasonable alternative for assigning a value to the hyperparameter based on the cosine similarities of the relation instances. Through experimentation across various entity pairs and embeddings, we can recommend that the value of the threshold should be in the vicinity of the third quartile of the cosine similarities of all pairs of relation instances. To illustrate the same, we have presented the v measure value across different entity pairs for flair based embeddings as well as infersent based embeddings. The third quartile value of the similarities of instances for flair based embeddings is 0.841,0.770 and 0.871 for person-degree, person-location, and person-organization entity pairs respectively, whereas for infersent based embeddings are 0.78,0.72,0.80 for person-degree, person-location, person-organization respectively. The threshold values in the figures 2 and 3 are presented in terms of degrees of the angle between the corresponding vectors to provide a better visualization. If you observe the figures figure 2 and figure 3 we can see that the difference of v measure for threshold values corresponding to 30 degrees(0.866) and 45 degrees(0.707) is much more for flair based embeddings compared to the infersent based embeddings. The reason being for the third quartile values for flair based embeddings, the threshold value corresponding to 30 degrees(0.866) is more in the vicinity compared to the threshold value corresponding to 45 degrees(0.707). In contrast, the third quartile values for infersent based embeddings are almost equally in the vicinity of both threshold values of 30 degrees (0.866) and 45 degrees (0.707). Although the v measure values seem to follow a similar trend for both the embeddings with an exception, which emphasizes that if the feature space captures the relation instance essence, the overall result would be similar irrespective of particular embedding models. So the current approach works well for a reasonable word embedding model, and more sophisticated embedding models are not required.

In figures 4 and 5 we illustrate the v measure, homogeneity, completeness and number of clusters information for comparison to the baseline. The current approach outperforms the baseline by huge margins across all the measures. The comprehensive metric, v measure, increased by multiple folds for all the entity pairs. The cost of labeling samples reduces as a result of the decline in the number of clusters. Despite the decrease in the number of clusters, the homogeneity scores have increased with corresponding significant increases in completeness score. Overall, the new approach outperformed

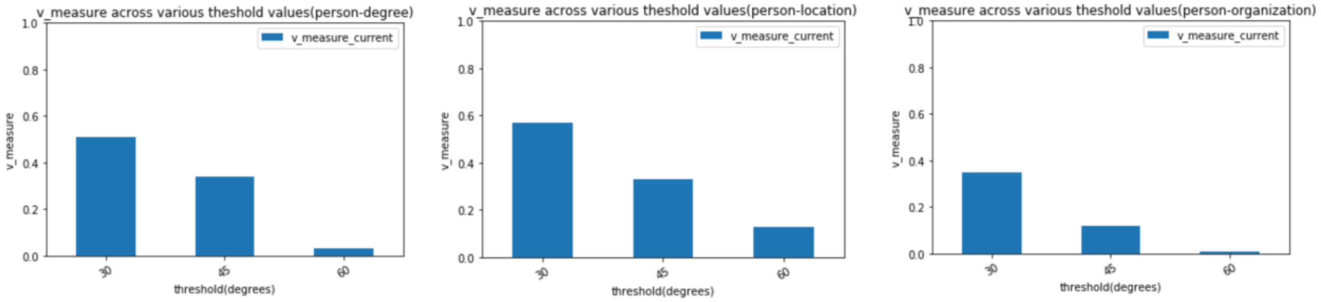

Fig. 2. Comparison of V measure for flair based embeddings

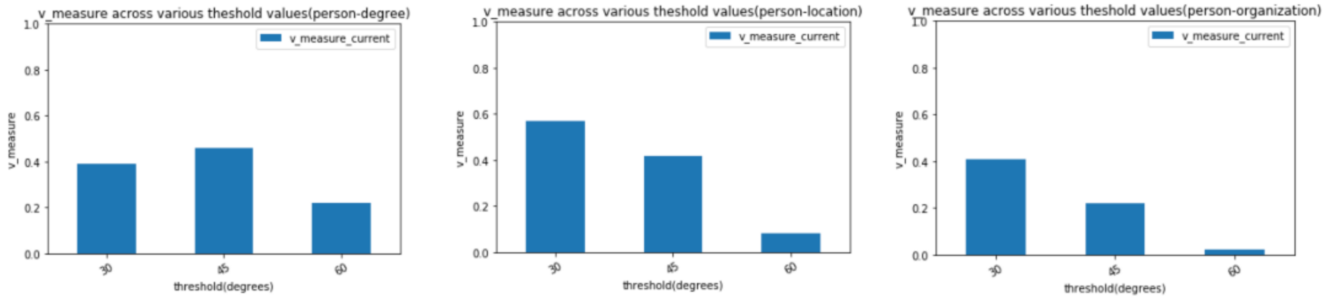

Fig. 3. Comparison of V measure for infersent based embeddings

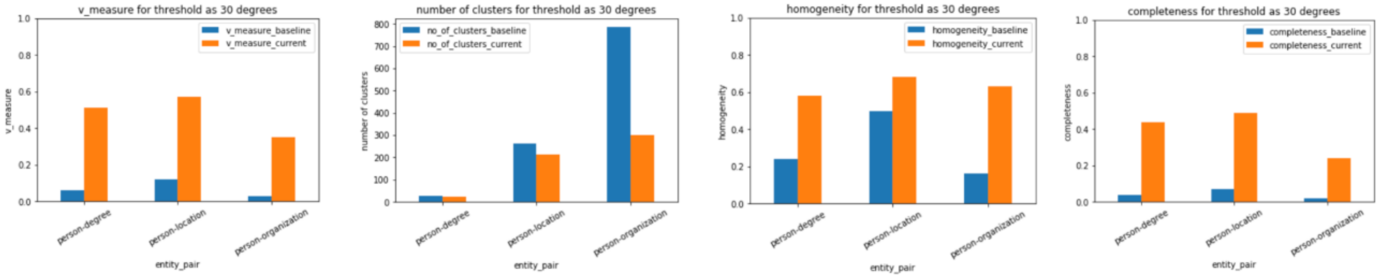

Fig. 4. Flair based embeddings [21] evaluation for threshold distance of 30 degrees

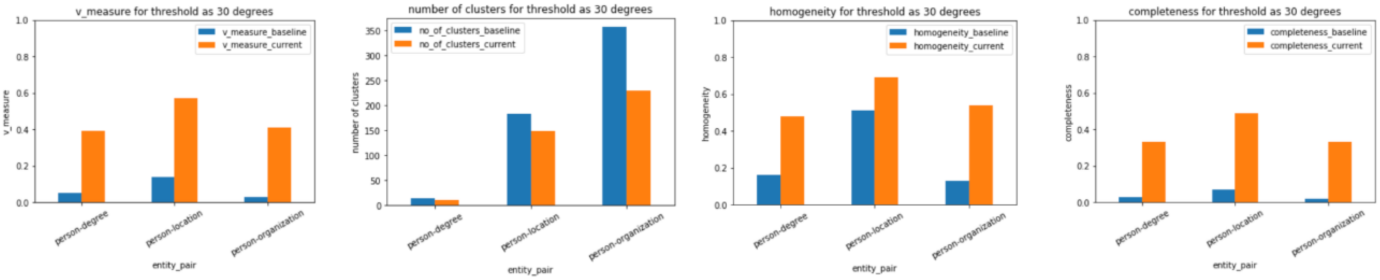

Fig. 5. Infersent based embeddings [19] evaluation for threshold distance of 30 degrees

the baseline across all metrics considered for features based on both flair [21] and infersent [19] models.

V. Conclusion and Future Work

In this paper, we have presented a novel hierarchical multi-level clustering algorithm to automatically label textual Big Data. The algorithm first divides data into different clusters based on semantic knowledge. Next each cluster is subject to Affinity Propagation based on the semantic vector representation of corresponding instances, followed by the reduction in the number of clusters with a novel approach to reassign instances to different clusters based on the similarity between them and potential cluster centroids. Finally, we sample clusters to generate a diverse set of samples, which comprises instances ranging from the most dissimilar to most similar with the cluster centroid. The Experts label these instances. A frequency-based approach then determines the appropriate label for each cluster. Our algorithm overcomes the need for existing knowledge bases or manual annotation to develop custom labeling functions.

We plan to extend this work by incorporating the current approach into a joint entity relation extraction framework with more research focus on entity extraction. We can also extend the algorithm by adopting other NLP tasks such as sentiment analysis or any other similar task by adapting the framework with specific changes to feature space.

- Varying Feature Space based Extension: We have discussed different features that can be extracted in the Background knowledge section but have opted for only entity pair and embedding features. We can change the features used by using a pair of categorical and numerical features, for instance, a different set of combinations of the features presented in an earlier chapter that captures and express the relation better. We can also increase the number of clustering levels for more fine-grained clustering.
- Adoption of Core Algorithm for Mainstream Clustering: Further exploration of the core clustering algorithm could result in a generic clustering algorithm, which is valid for a diverse set of tasks and applies to various domains besides natural language processing.

References

[1] Ratner, Alexander, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. ”Snorkel: Rapid training data creation with weak supervision.” The VLDB Journal (2019): 1-22.

[2] Brin, Sergey. ”Extracting patterns and relations from the world wide web.” International Workshop on The World Wide Web and Databases. Springer, Berlin, Heidelberg, 1998.

[3] Agichtein, Eugene, and Luis Gravano. ”Snowball: Extracting relations from large plain-text collections.” Proceedings of the fifth ACM conference on Digital libraries. 2000.

[4] Hasegawa, Takaaki, Satoshi Sekine, and Ralph Grishman. ”Discovering relations among named entities from large corpora.” Proceedings of the 42nd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2004.

[5] Chen, Jinxiu, et al. ”Unsupervised feature selection for relation extraction.” Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts. 2005.

[6] Yan, Yulan, et al. ”Unsupervised relation extraction by mining wikipedia texts using information from the web.” 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 2-Volume 2. Association for Computational Linguistics, 2009.

[7] Yao, Limin, et al. ”Structured relation discovery using generative models.” Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[8] Mintz, Mike, et al. ”Distant supervision for relation extraction without labeled data.” 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 2-Volume 2. Association for Computational Linguistics, 2009.

[9] Bollacker, Kurt, et al. ”Freebase: a collaboratively created graph database for structuring human knowledge.” Proceedings of the 2008 ACM SIGMOD international conference on Management of data. 2008.

[10] Auer, Sören, et al. ”Dbpedia: A nucleus for a web of open data.” The semantic web. Springer, Berlin, Heidelberg, 2007. 722-735.

[11] Orr, Dave. ”50,000 Lessons on How to Read: a relation extraction corpus.” Online: Google Research Blog 11 (2013).

[12] Mikolov, Tomas, et al. ”Distributed representations of words and phrases and their compositionality.” Advances in neural information processing systems. 2013.

[13] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[14] Bojanowski, Piotr, et al. ”Enriching word vectors with subword information.” Transactions of the Association for Computational Linguistics 5 (2017): 135-146.

[15] Moody, Christopher E. ”Mixing dirichlet topic models and word embeddings to make lda2vec.” arXiv preprint arXiv:1605.02019 (2016).

[16] Peters, Matthew E., et al. ”Deep contextualized word representations.” arXiv preprint arXiv:1802.05365 (2018).

[17] Radford, Alec, et al. ”Improving language understanding by generative pre-training.” URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018).

[18] Bowman, Samuel R., et al. ”A large annotated corpus for learning natural language inference.” arXiv preprint arXiv:1508.05326 (2015).

[19] Conneau, Alexis, et al. ”Supervised learning of universal sentence representations from natural language inference data.” arXiv preprint arXiv:1705.02364 (2017).

[20] Ester, Martin, et al. ”Density-based spatial clustering of applications with noise.” Int. Conf. Knowledge Discovery and Data Mining. Vol. 240. 1996.

[21] Akbik, Alan, Duncan Blythe, and Roland Vollgraf. ”Contextual string embeddings for sequence labeling.” Proceedings of the 27th International Conference on Computational Linguistics. 2018.

[22] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. ”CURE: an efficient clustering algorithm for large databases.” ACM Sigmod record 27.2 (1998): 73-84.

[23] Wu, Shanchan, Kai Fan, and Qiong Zhang. ”Improving Distantly Supervised Relation Extraction with Neural Noise Converter and Conditional Optimal Selector.” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7273-7280. 2019.

[24] Raiman, Jonathan Raphael, and Olivier Michel Raiman. ”Deeptype: multilingual entity linking by neural type system evolution.” In Thirty-Second AAAI Conference on Artificial Intelligence. 2018.