IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A COMPREHENSIVE REVIEW OF BOOTSTRAPPING PATTERN LEARNING TECHNIQUES IN INFORMATION EXTRACTION

[1]Manisha Mali, [2]Dr. Neeraj Sharma

[1]Research Scholar, [2]Associate Professor

[1,2]Dept. of CSE, SSSUTMS, Sehore, MP, India

Abstract: The automatic extraction of information from unstructured sources has opened up new avenues for querying, organizing, and analyzing data by drawing upon the clean semantics of structured databases and the abundance of unstructured data. The field of information extraction has its genesis in the natural language processing community. The primary impetus came from competitions centered around recognizing named entities like people names and organizations from news articles. This paper surveys Bootstrapping Pattern Learning Techniques in Information Extraction and presents detailed talk of the algorithms andthe relative pros and cons of the techniques used. We also study theproblems various authors attempt to solve and analyze when it is appropriate to utilize a bootstrapping algorithm.

Index Terms - **Information Extraction, Natural Language Processing, Bootstrapping Pattern Learning**.

I. INTRODUCTION

A considerable amount of digital data is available on the internet and intranets due to so many reasons. First, a significant part of digital information is transmitted through free-text documents, i.e., unstructured, and is thus difficult to search in. Data is generated by government documents, corporate reports, online news, court rulings, legal acts, medical alerts and records, and social media communications. This resulted in an increasing need for effective and efficient techniques for analyzing free-text data and discovering valuable and relevant knowledge from it in structured information and led to the emergence of Information Extraction technologies. Statistical approaches to information extraction and natural language processing require vastamounts of data to perform acceptably and produce reliable results. Likewise, empirical training algorithms require an extensive database. A corpus of uninterpreted, unmarked natural language text is not often suitable for most tasks; algorithms typically need their training data to be annotated in some fashion to extract and learn the salient textual features.

Unfortunately, annotated databases are in short supply. Manually annotating even a small corpusis incredibly time-consuming and is infeasible forlarger ones. Furthermore, automatically tagging a corpus with the necessary information is usually not possiblesince the annotations needed for training are typically the information we are trying to find in the first place. For example, a practical algorithm that attempts to identify whether a given article is written subjectively or objectively in fashion would typically train on a corpus marked up with annotations indicating subjectivity and objectivity in some manner. The algorithm would then learn from this annotated training corpus and then(hopefully!) effectively perform this task on unmarked natural language text.

Bootstrapping provides an alternative to precise manual annotation. The techniques reviewed herein are all trained on unmarked corpora. The implementers provide their algorithm with a small number of carefully chosen seeds. These seeds are then used as a starting point to gather other, similar terms from an unmarked training corpus, which are in turn then used to gather even more terms, and so on. In essence, the algorithm "pulls itself up from its bootstraps," hence the name.

The rest of the paper is organized as follows. Section II covers information extraction. Section III focuses on a Bootstrapped learning algorithm. A literature survey of work done in the field of information extraction and Bootstrapped learning algorithm, along with the background, is presented in section IV. Performance evaluation in information extraction is shown in section V. Finally, the conclusion and possible directions for future research have discoursed.

II. INFORMATION EXTRACTION

The process of information extraction is divided into different steps of various granularity by various researchers. However, analysing those different approaches to prepare the broad pipeline of the information extraction process is done here. In this, six main stages are determined as follows: 1. Preprocessing. 2. Proper name identification. 3. Parsing. 4. Extraction of events and relations. 5. Coreference resolution. 6. Output results generation. [Varsha Pande et al., 2016]

1 Pre-processing: There are several operations involved in the primary step of the information extraction preprocessing. The first is splitting a text into fragments defined differently throughout the papers from different researchers like zones, sentences, segments, or tokens. This procedure can be performed by the components named tokenizers, text zoners, segmenters, or splitters. Tokenization is a pretty straightforward task for texts in any language. The blank space between characters and punctuation indicates the

boundaries of a word and a sentence, respectively (separation of tokens). The next task within the initial processing stage is usually the morphological analysis which includes part-of-speech tagging and phrasal units (noun or verb phrases) identification. Part-of-speech tagging might be helpful to the next step, which is the lexical analysis. It deals with unknown words and resolves ambiguities by identifying part-of-speech of the words that cause those ambiguities. Also, the lexical analysis involves working with the specialized dictionaries and gazetteers, which are composed of different types of names: titles, countries, cities, companies and their suffixes, positions in a company, etc. If a word in a document is found in a gazetteer, it is tagged with the semantic class the word belongs to. For example, the word "Mr" will be tagged with the semantic class "Titles." Some authors add a filtering task to the preprocessing stage, which implies selecting only those sentences which are relevant to the extraction requirements.

2 Proper name identification: One of the essential steps in information extraction is recognizing various classes of proper names, such as names of people or organizations, dates, currency amounts, locations, addresses, etc. They can become across in almost all types of texts, and usually, they constitute part of the extraction scenario. These names are recognized using several patterns called regular expressions[Feldman, R., 2007].

3 Parsing: During this stage, the syntactic analysis of the sentences in the documents is performed. This parsing stage must be done to prepare the ground for the next step, i.e., extraction of events and relations, between those entities and circumstances in which they participate. The noun and verb groups are used as sections to begin to work on at the pattern matching stage. The identification of those groups is realized by applying a set of specially constructed regular expressions. Performing syntactic analysis on the whole document is difficult, so information extraction research groups tend to use so-called partial or shallow parsing instead of the full one. The shallow parsing creates partial, not overlapping syntactic fragments identified with a higher confidence level using only local information. [Grishman, R., 1997].

4 Extraction of events and relations: Output produced by the previous three steps is preparing for the significant stage of extraction of events and associations, particularly related to the preliminary extraction specifications given by a client. This process is realized by creating and applying extraction rules which specify different patterns. Then, the text is matched against those patterns, and if a match is found, the text element is labeled and later extracted. The formalism of writing those extraction rules differs from one information extraction system to another. [Douglas E. Appelt, et. al., 1999].

5 Coreference resolution: Any given entity in a text can be referred to several times, and every time it might be referred to differently. To identify all the ways used to name that entity throughout the document, coreference resolution is performed. Coreference or anaphora resolution is the stage when for noun phrases it is determined if they refer to the same entity or not. There are several types of coreference, but the most common types are pronominal and proper names coreference when a noun is replaced by a pronoun in the first case and by another noun or a noun phrase in the second one[Feldman, R., 2007].

6 Output results generation: This step involves transforming the structures extracted during the previous operations into the output templates according to the format specified by a client. It might include different normalization operations for dates, time, currencies, etc. For instance, a round-off procedure for percentages can be executed, and an actual number 75.96 will be turned into integer 76[J Turmo et al., 2006]. Of course, a particular information extraction project or system does not have all of these components. Several factors affect the selection of systems' features, like:

- **Language**: As mentioned earlier, for processing texts in Chinese or Japanese languages with no clear word and sentence boundaries or texts in the German language with words of a complicated morphological structure, some modules are necessary compared to working with English documents.
- **Text genre and properties:** In transcripts of informal speech, for example, spelling mistakes might occur in addition to implicit sentence boundaries. If information must be extracted from such texts, those issues must be considered and addressed while designing a system by adding corresponding modules.
- **Extraction task:** For an easy task like name recognition, the parsing and Coreference resolution modules might not be needed at all.

III. BOOTSTRAPPING ALGORITHMS

Bootstrapped Pattern Learning is used for learning patterns to learn entities of given entity types from unlabeled text, starting with seed sets of entities. Though there is significant variation between implementations, bootstrapping approaches in natural language processing all follow the same general format:

1. Start with an empty list of things. Usually words or phrases, but it can be any representation of language (such as regular expressions, tuples, etc.)
2. Initialize this list with carefully chosen seeds (the initial set of data).
3. Leverage the things in the list to find more items from a training corpus.
4. Score those newly found things; add the bestones to the list.
5. Repeat step 3. Stop after a set number of iterations or some other stop condition. The intuition for bootstrapping stems from the observation that words from the semantic category tend to appear in similar patterns and similar contexts. For example, the words "water" and "soda" are both in the semantic BEVERAGE category, and in a largetext, they will likely both be found in phrasescontaining drank or imbibed. The core intuition is that by searching

The core feature of a bootstrapping algorithm isthat each iteration is fed the same type of data as its input that it produces as its output. The outputof the first iteration is used as the input of the second iteration, and so on. It should not be surprising that choosing the initial set of data (the seeds)from which all other data is "grown" is a critical factor (arguably the most critical factor) in the performance of the algorithm.[Daniel Waegel] At their core, bootstrapping algorithms are not limited to learning general semantic categories. They help understand any variety or relationship for which words appear in similar linguistic phrases or contexts.

Weaknesses of Bootstrapping include Semantic Drift and Stop conditions. A very well-known flaw in bootstrapping is a phenomenon known as semantic drift or creep. This occurs when, after multiple iterations, the bootstrapper wanders away from the original semanticmeaning of the seeds and begins to accept incorrect or undesirable entities. This can occur if entities have more than one semantic word sense, it can also happen if words similar meaning and usage buthave different undertones. Another major drawback that was bootstrapping suffers isthe difficulty in knowing when to stop the iterativecycles. Some quit after a set and seemingly arbitrary number of iterations, while others [Lin et al., 2003] stopped when theiralgorithm had exhaustively added all candidates or rejected any remaining candidates. Finally, the choosing of seeds is arguably the most critical step in bootstrapping. Most authors

chose the most frequently occurring words in their corpus that they have quickly identified belong to the category they are interested in. While this ensures that the most outstanding amount of contextual information will be available to learn from, it does nothing to ensure the quality of the contexts. It is easy to imagine improperly chosen seeds that would pick up tons of extraction patterns that, in turn, extract feeble additional words, ultimately producing poor results.

IV. LITERATURE SURVEY

Agichtein et al., (2000) outlines an attempt to identify and extract structuredrelations between named entities from unstructured text. In this context, a relation is defined as a table or listing that maps one entity onto another (as in a relational database). The example used throughout the article is mapping an organization entity (e.g., Microsoft or Boeing) onto their headquarters (a location entity such as Redmond for Microsoft). These are represented as a tuple. This research aims toallow much more nuanced responses to questionssuch as "Where is Microsoft headquartered?". They name this system Snowball. Snowball is built on top of the DIPRE (Dual Iterative Pattern Relation Expansion) method. The central idea behind both of theseexploits the idea, termed pattern relation duality,that a "good" extraction pattern will produce goodtuples given a large enough body of text, and a "good" set of tuples can have good extraction patterns deduced from it. The only user-provided input to this Snowball system is the training corpus and a small handful of manually compiled relations (the seeds).

Michael Thelen et al. (2002) described a bootstrapping algorithm called Basilisk ((Bootstrapping Approach to SemantIc Lexicon Induction using Semantic Knowledge)) that learns high-quality semantic lexicons for multiple categories. Basilisk begins with an unannotated corpus and seed words for each semantic category, which are then bootstrapped to learn new words for each category. Basilisk hypothesizes the semantic class based on collective information over a large body of extraction pattern contexts. They utilize a bootstrapping approach to developing a semantic lexicon of nouns. This is a categorization task: Search for nouns in an unannotated corpus and assign them to one of six semantic categories. These semantic categories are BUILDING, EVENT, HUMAN, LOCATION, TIME, and WEAPON. The authors manually selected these ad hoc categories because the corpus comprises articles written about terrorism. Basilisk, like all bootstrapping algorithms, must be seeded with carefully selected terms tobe effective. They seeded theiralgorithm by sorting the words in the corpus by frequency and manually identifying the 10 most frequent words for each of the six semantic categories. Then, they evaluated Basilisk on six semantic categories. Basilisk's bootstrapping algorithm exploits two ideas: (1) collective evidence from extraction patterns can be used to infer semantic category associations, and (2) learning multiple semantic categories simultaneously can help constrain the bootstrapping process. The semantic lexicons produced by Basilisk have higher precision than those produced by previous techniques, with several categories showing substantial improvement.

[Thelen and Riloff, 2002] and [Lin et al. 2003] use the NOMEN algorithm, although the goals of each paper were very different. The NOMAN algorithm was first formally described in the literature by [Yangarber et al.,2002],where it is used to address the problem of learninggeneralized names in a biomedical context. Generalized words, such as mad cow disease or Ebolahemorrhagic fever, often lack the (whole) capitalization cues afforded to proper names. Therefore, detection methods must rely on other metrics to attain good performance. The authors mainly focus on the confounding factor of semantic ambiguity in the bootstrapping process. It is common fornames in the biomedical context to refer to botha disease and a symptom, resulting in ambiguity and significantly increasing the difficulty of distinguishing between the two semantic classes. The NOMEN algorithm requires its training setto undergo significant preprocessing: The training corpus is run through a zoner, a sentence splitter, a tokenizer/lemmatizer, and a part-of-speech tagger. The zoner must extract the actualcontent because the training corpus collects mailing list correspondence with mailing headers and footers. The splitter and tokenizer break the corpus into word tokens; the lemmatizer converts words to their base form, which are finally tagged by the part-of-speech tagger.

[Riloff and Wiebe, 2003] uses bootstrapping to identify subjective expressions in the system named AutoSlog-TS. Empirical methods have previously been leveraged to create listsof words and n-grams statistically associated withsubjective language. However, subjective ideas are expressed in myriad forms, many of which are very hard to detect using conventional natural language processing approaches. For example, sarcastic, metaphorical, and idiomatic phrases are cases where empirical methods easily overlook the expression of subjectivity; both utilize words that may be objective when isolated but may become intensely subjective when placed in context. The Autoslog-TS algorithm input the set of relevant (subjective) and irrelevant (objective) sentences. First, syntactic templates generated from prior work are exhaustively applied to the training corpus. Then, extraction patterns are generated for every possible instantiation of the templates that appear in the training corpus.

Bootstrapped pattern learning for entity extraction usually starts with seed entities and iteratively learns patterns and entities from unlabeled text. Patterns are scored by their ability to extract more positive entities and less negative entities. Due to the lack of labeled data, unlabeled entities are either assumed to be negative or are ignored by the existing pattern scoring measures. Sonal Gupta et al. (2014) improve pattern scoring by predicting the labels of unlabeled entities. They use various unsupervised features based on contrasting domain-specific and general text, exploiting distributional similarity and editing distance to learned entities. Thus they show that predicting the labels of unlabeled entities in the pattern scorer of a bootstrapped entity extraction system significantly improves precision and recall of known entities. Their experiments demonstrate the importance of having models that contrast domain-specific and general domain text and the usefulness of features that allow spelling variations when dealing with informal texts. In this system, pattern scorer outperforms existing pattern scoring methods for learning drug and treatment entities from four medical web forums.

Biomedical NER, i.e., biomedical named entity recognition, is a core component to build biomedical text processing systems, such as biomedical information retrieval and question answering systems. The machine learning-based approaches generally require significant amounts of annotated corpora to achieve high performance. However, it is expensive to manually create many high-quality corpora due to the demand for biomedical experts. Also, most existing corpora have focused on many specific sub-domains, such as disease, protein, and species. Therefore, it is difficult for a biomedical NER system trained with these corpora to provide much information for biomedical text processing systems. [Juae Kim et al.,2019] proposed a method for automatically generating the machine-labeled biomedical NER corpus covering various sub-domains by using proper categories from the semantic groups of a united medical language system (UMLS). They use a bootstrapping approach with a small amount of manually annotated corpus to automatically generate a significant amount of corpus and then construct a biomedical NER system trained with the machine-labeled corpus. In the end, They train two machine learning-based classifiers, conditional random fields (CRFs) and long short-term

memory (LSTM), with the machine-labeled data to improve performance. As shown by experimental results, the proposed method is effective in improving performance. As a result, the proposed one obtains higher performance in 23.69% than the model that trained only a small amount of manually annotated corpus in F1-score.

V. PERFORMANCE EVALUATION IN INFORMATION EXTRACTION

Given an input text or a collection of texts, the expected output of an IE system can be defined very precisely. This facilitates the evaluation of different IE systems and approaches. In particular, the precision and recall metrics were adopted from the Information Retrieval research community for that purpose. However, the system's effectiveness is measure from the user's perspective, i.e., the extent to which the system produces all the appropriate output (recall) and only the appropriate output (precision). Thus, recall and precision can be seen as a measure of completeness and correctness, respectively. While precision and recall have served the information extraction community well as two separate measures of system performance, but the F-measure, the weighted harmonic mean of precision and recall, exhibits certain undesirable behaviors. To overcome these limitations, [John Makhoul et al.] defines an error measure, the slot error rate (SER), which combines the different types of the error directly without resorting to precision and recall as preliminary measures. The slot error rate is analogous to the word error rate used for measuring speech recognition performance; it is intended to calculate the cost to the user for the system to make the different types of errors. Slot Error Rate is simply the ratio of the total number of slot errors – substitutions, deletions, and insertions – divided by the total number of slots in the reference, fixed for a given test. In this way, the errors from all systems are compared against a fixed base.

VI. CONCLUSION

We have studied in detail various approaches to bootstrapping for tasks involved in natural language processing. After reviewing so many research papers, we conclude that bootstrapping is a powerful method for extracting valuable data from enormousvolumes of unstructured text, which will help perform tasks in information extraction. Bootstrapping is responsive to various information extraction tasks, such as relation extraction, named entity recognition, semantic categorization, and subjectivity analysis. Though, bootstrapping systems must be carefully seeded and constrained to avoid growing in unwanted directions or becoming excessively diluted with irrelevant data. The choice of seeds is pivotal to the success of thebootstrapping process, and it is not at all clear howto determine what the "best" seed might be. Bootstrapping systems are well suited to natural language tasks due to their ability to learn and navigate the syntactically rich, unstructured, and highly complex nature of loosely structured natural languages. The factors such as precision, recall, F-measure, and slot error rate are used to evaluate information extraction.

REFERENCES

[1] Agichtein, Eugene and Gravano, Luis. 2000. Snowball: Extracting relations from large plain-text collections. In Proceedings of the fifth ACM conference on Digital libraries, pages 85–94. ACM.

[2] Douglas E. Appelt and David J. Israel, 1999. Introduction to Information Extraction Technology. IJCAI-99.

[3]Feldman, R., 2007. The Text Mining Handbook: Advanced Approaches In Analyzing Unstructured Data. New York, Cambridge University Press

[4] Grishman, R., 1997. Information Extraction: Techniques and Challenges. Berlin, Heidelberg, Springer-Verlag, pp. 10-27.

[5] J Turmo, et al., 2006. Adaptive Information Extraction. ACM computing surveys, pp. 1-47.

[6] John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel, "Performance Measures For Information Extraction"

[7] Juae Kim, Youngjoong Ko, Jungyun Seo, "A Bootstrapping Approach With CRF and Deep Learning Models for Improving the Biomedical Named Entity Recognition in Multi-Domains," 2169-3536 2019 IEEE Translations and content mining

[8] Lin, Winston, Yangarber, Roman, and Grishman, Ralph. 2003. Bootstrapped learning of semantic classes from positive and negative examples. In Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Volume 4, No. 4.

[9] Riloff, Ellen, Wiebe, Janyce and Wilson, Theresa. 2003. Learning subjective nouns using extraction pattern bootstrapping. In Proceedings of the seventh conference on Natural language learning at HLTNAACL, Volume 4, pages 25–32. Association for Computational Linguistics.

[10] Sonal Gupta Christopher D. Manning, "Improved Pattern Learning for Bootstrapped Entity Extraction," Proceedings of the Eighteenth Conference on Computational Language Learning, pages 98–108, Baltimore, Maryland USA, June 26-27 2014. c 2014 Association for Computational Linguistics

[11] Thelen, Michael and Riloff, Ellen. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Volume 10, pages 214–221 Association for Computational Linguistics.

[12] Varsha Pande, Dr. A. S. Khandelwal, "Information Extraction Technique: A Review," IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727 PP 16-20

[13] Yangarber, Roman, Lin, Winston, and Grishman, Ralph. 2002. Unsupervised learning of generalized names. In Proceedings of the 19th international conference on Computational linguistics, Volume 1.Association for Computational Linguistics