Automatic Extraction of Fact Using Formal Digital Content

Naman Sharma[1*], Nehul Sukralia[2], Nikhil Katiyar[3], Ritesh Srivastava[4]

[1,2,3]*Student, Department of Computer Science and Engineering Galgotias College of Engineering and Technology, Greater Noida, India*

[4]*Professor, Department of Computer Science and Engineering Galgotias College of Engineering and Technology, Greater Noida, India*

*Corresponding author: naman.1998sharma@gmail.com

***Abstract*: The process of extracting disambiguated data from natural language texts in service of some pre-specified precise information is called fact extraction. The modern era has seen a huge development of Internet technologies that has followed to information explosion in past 10 years. This change can be seen by an intensive growth of data volume among the low-quality data. Fact extraction is the process of automatically extracting structured information from unstructured and/or semi-structured machine-readable document. It is the analysis of natural language in order to extract snippets of knowledge from information. This survey paper brief about studies or approaches which have been used in on-going project of developing fact extraction technology on Textual data available in any form.**

***Keywords*: Structured information, Readable document, Textual data.**

1. Introduction

Automatic Fact Extraction is a very important aspect in the field of Natural Language Processing and linguistics. It is being largely used in automating tasks such as Question Answering Systems, Entity Extraction, Event Extraction, Named Entity Linking, Relation Extraction, etc.

There is an important concept of triples information extraction. A triple represents a combination of entities and an existing relation between them. For example, (Delhi, capital, India) is a triple in which ‘Delhi’ and ‘India’ are the related entities, and the relation between them is ‘capital’.

Fact discovery describes the process of automatically searching large amount of data for patterns that can be considered knowledge about the data. With tons of individual pages on the web providing information about every possible topic, we should be able to collect facts that answer every possible question.

The process of finding takes texts data as input and produces specified format, unambiguous data as output

Fact finding is developed out of the data mining domain, and is closely associated with it in both terms of methodology and terminology.

Basically there are two approaches of information extraction:

In traditional way of fact extraction, the relationship to be extracted are pre-defined.

In Open Information Extraction, the relationship is not pre-defined. The system is boundless to extract any relations it comes across while going through the text data.

Information Extraction is quite different from Information Retrieval:

- The IR system finds relevant texts and presents them to the user;
- The IE application analyses texts and presents only the specific information from them that the user is interested in [1].

2. Literature Survey

A. *Briefing*

A fact extraction tool kit finds and extracts desired pieces of information from textual data using linguistic and pattern matching technology, and specifically, text description and fact extraction. Text annotation tools divide a text, such as docs, into its base tokens and specify those tokens or patterns of tokens with, syntactic, semantic, pragmatic attribute [2].

B. *Categorization*

The sentences spoken in the speeches are divided into three categories:

Non-Factual Sentence (NFS): Like subjective sentences (opinions, declarations, beliefs) and many queries fall under this category.

These sentences do not contain any fact based claims. Below are two such examples.

- But I think you should buy a car.
- You remember the last time we met?

Unimportant Factual Sentence (UFS): These are factual claims but not important. One will not be interested in knowing whether these sentences are false or true. Fact-checking mechanism would not find these sentences important for checking. Some examples are as follows.

- Your birthday is on upcoming Saturday.
- We went to picnic yesterday.

Check-worthy Factual Sentence (CFS): They contain factual claims and one will be interested in knowing whether the assertions are true. Journalists seek for these type of assertions for fact-checking. Some examples are:

- Your heart pumps blood through your body.
- There are 50 states in the United States [3].

C. *Need of Fact Extraction*

Information comes in many shapes and sizes. One important form is structured data, where there is a regular organization of entities and relationships.

Information extraction is the task of extracting structured information from unstructured and/or semi-structured machine-readable documents [4].

The present significance of IE pertains to the growing amount of information available in unstructured format.

The Further analysis of extracted text to know the amount of factual information present is called Fact Extraction [5].

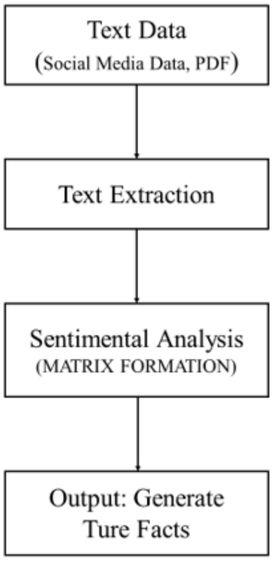

Fig. 1. Flow of extraction processing

D. *Problem Statement*

The survey tells about an idea which aims at solving the problem of information extraction integrated with factual analysis of extracted data.

And deciding on the terms of analysis whether the data is point to point or of General Information.

This aims at helping the user in selection of digital content depending on his/her prior knowledge and Interest. The Fig. 1 shows the Flow of steps for Fact Extraction.

Different Approaches to Information Extraction

In Traditional Information Extraction, the relations to be extracted are predetermined. In this content, we will cover the rule-based methods only.

In open information extraction, the relations are not predetermined. The system is free to extract any relations it comes across while going through the data [6].

E. *Approaches to perform information extraction*

1. Rule-based Approach: A set of rules are defined for syntax and other grammatical properties of a natural language and then use these rules to extract factual information from text.
2. Supervised: Suppose we have a Phrase P. It has two entities X1 and X2. The supervised ML model has to detect whether there is any relation (R) between X1 and X2. Therefore, in a supervised approach method, the idea of relation extraction turns into the idea of relation detection. The only disadvantage of this method is that it needs a lot of labelled data for training purpose.
3. Semi-supervised: When we don’t have sufficient labelled data, we can use a set of pre-defined examples (triples) to create more accurate patterns that can be used to extract more similar relations from the text [7].

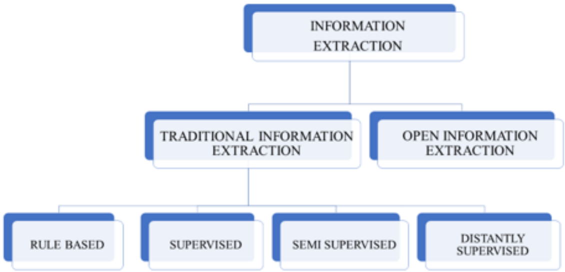

Fig. 2. Various approaches of extraction

3. Related Approaches

Carnegie Group recently developed and deployed a fact extraction system called JASPER for Reuters Ltd. Template based driven approach and partial understanding techniques to extract specific key pieces of information from a constrained range of text is used by JASPER. Predetermined set of information is used by JASPER to extract information.

Excellent results in terms of accuracy and speed is obtained by JASPER. This is achieved by combining frame-based object-oriented processing, knowledge representation, powerful pattern matching, and heuristics are used which take advantage of stylistic conventions, including lexical, semantic, syntactic, and irregular pragmatic characteristics are observed in the text.

A. *Technical Approach*

JASPER first determines whether it is relevant or not that whether it is one of the earning or extra releases from which we wish to extract information.

Specific information types to be extracted from relevant texts are defined in JASPER. Frame representation system processing is guided by the remaining Frames.

Subsequent outcome of this approach has shown that JASPER does its job with accuracy and rapidly.

- It tooks approximately 25 seconds for JASPER to processes the average earnings or dividend release.

- The accuracy of the system is 96% in selecting relevant releases for processing.
- By corresponding measures for The factual measure of extraction system is over 84% accurate in extracting the desired knowledge from the selected releases.
- Values placed by JASPER in their stories is about 90% correct.
- 33% of targeted releases perfectly handled by JASPER. Earning stories with approximately no errors or omission are handled by somewhat 21 % and handles about 82% of all dividend releases with least error or omissions [8].

B. *Syntactic Analysis in Information Extraction Systems*

It is a question of big compromise in Natural language processing in information extraction systems is often making compromises to what one might consider ideal natural-language processing, associated by the need to process large quantities of real-world text within specific time constraints.

Semantic analysis is the process of relating syntactic structures in Linguistics from the different levels of phrases, clauses, sentences and paragraphs to the mark of the writing as a complete description, to their language-independent meanings. It involves removing particular linguistic and cultural contents to the extent that such a project is possible.

Elements such as idiom and figurative speech, being cultural, are usually also converted into relatively invariant meanings in semantic analysis. Semantics, although related to pragmatics, is specific in that the former deals with word or sentence choice in any given context, while pragmatics considers the unique or particular meaning derived from context or tone. To reiterate in different phrase, semantics is about universally understandable meaning, and pragmatics, the meaning encoded in words that is then interpreted by a receiver or subject [9].

C. *Semantic Analysis in Information Extraction Systems*

Tokenization is the process of tokenizing or break a string, text into a list of tokens. One can think of token as different parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

Lexical analysis is the process of converting an input stream of characters into a stream of tokens. Tokens are basically groups of characters with collective significance in any text. Lexical analysis is the first phase of automatic indexing, and of query processing.

Automatic indexing is the technique of algorithmically examining information items to generate lists of index terms. The lexical analysis phase produces candidate index terms that may be further processed, and finally added to indexes. Lexical analysis of a query produces tokens that are parsed and turned into an internal representation suitable for comparison with indexes [10].

Key points of the article,

- Text into sentences tokenization
- Sentences into words tokenization
- entences using regular expressions tokenization

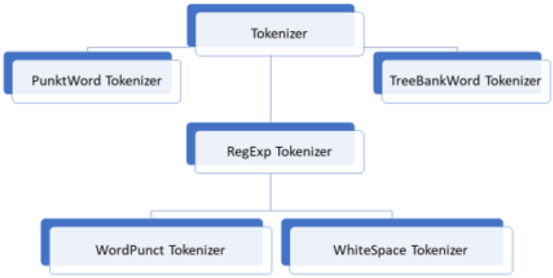

Fig. 3. Tokenization

1) *Tokenization*

Further processing is generally done after a piece of text has been properly tokenized. As stated and shown in Fig. 3, that Tokenization is also referred to as text segmentation or lexical analysis. The breakdown of a large chunk of text into pieces larger than words are sometimes referred through segmentation (e.g. paragraphs or sentences), while tokenization is specifically meaning the breakdown process which results in words.

This may sound like a straightforward process, but it is not. Sentences are identified within larger texts is through sentence-ending punctuation.

This sentence is easily identified with some basic segmentation rules:

- The quick brown fox jumps over the lazy dog.
- But what about this one:
- Dr. Cruise did not ask Col. David the name of Mr. Smith’s dog.
  Or this one:
- "What is all the fuss about?" asked Mr. James.

2) *Normalization*

Before moving forward, text needs to be normalized. Normalization generally refers to a series of related tasks meant to put all text on a same level converting all text to the same case (upper or lower), removing punctuation, converting numbers to their word format. Normalization puts all words on equal level, and allows processing to proceed swiftly, refer to Fig. 4.

Normalizing can be completed in three different steps:
(1) stemming, (2) lemmatization, and (3) everything else.

1. Stemming:

Stemming is the process of removing suffixed, prefixes, infixes, circumfixes from a word in order to obtain a word stem. eating → eat

2. Lemmatization:

Lemmatization is related to word stemming; it is to capture canonical forms based on a word's lemma.

For example, stemming the word "worst" would fail to return it’s another word for lemma; however, lemmatization would result in the following:

worst → bad

It should be easy to see why the implementation of a stemmer would be the less difficult part of the two [11].

3. Everything else:

Stemming and lemmatization are major parts of a text pre-processing endeavour. These are not any simple text manipulation; they rely on detailed and specific understanding of grammatical rules and norms.

However, multiple other steps can be taken to help put all text on equal level, many of which involve the comparatively simple ideas of replacement or removal. They are, moreover of no less importance to the overall process. These include:

- Set all characters to lowercase.
- Remove numbers or convert numbers to textual representations.
- Removing punctuation is general part of tokenization, but still worth keeping in mind at this stage, even as confirmation.
- Strip white space is also generally part of tokenization.
- Remove default stop words in general English.

Before further processing of text Stop words are filtered out, as these words contribute less to overall meaning, also they are generally the most common words in a language. For instance, "the," "and," and "a," basically all required words in a passage generally do not contribute greatly to one's understanding of the content. Just a simple example, the following pangram is just as readable if the stop words are removed:

The quick brown fox jumps over the lazy dog.

- Removing given is task-specific stop words.
- Removing sparse terms is not always necessary or helpful, though!

So, this is clear that text pre-processing relies a lot on pre-built dictionaries, databases, and rules [12].

3) Noise Removal

Substitution tasks of the framework helps in Noise removal. While the first 2 major steps of our framework such as tokenization and normalization were generally acceptable as is to nearly any text chunk or project (barring the decision of which exact implementation was to be employed, or skipping certain optional steps, such as sparse term removal, which simply does not apply to every project), noise removal is a much more task-specific section of the framework.

Keeping in mind that this is not a linear process, the steps of must exclusively be applied in a specified order. Noise removal, therefore, can occur before or after any steps.

Assuming that we have obtained a chunk from the world wide web, and that it is received as a raw web format. Then there are high chances that our text could be wrapped in HTML or XML tags. As this consideration for metadata can take place as part of the text collection or assembling process, it depends on how the data was gathered and combined.

But it's not necessary. If the chunk you have been using is noisy, you have to deal with it. The analysis tasks are often discussed about as being 80% data preparation!

The existing software tools built to deal with just such pattern matching tasks.

- Remove text file headers, footers.
- Remove HTML, XML, etc. mark up and metadata.
- Extract valuable data from other formats, such as JSON, or from within databases
- Regular expressions, this could majorly the part of text pre-processing in which your worst fears are realized.

The boundary between noise removal and data collection and assembly is a fuzzy one, and as such some noise removal must take place before other pre-processing steps. For example, any text required from a JSON structure would obviously need to be removed prior to tokenization [13].

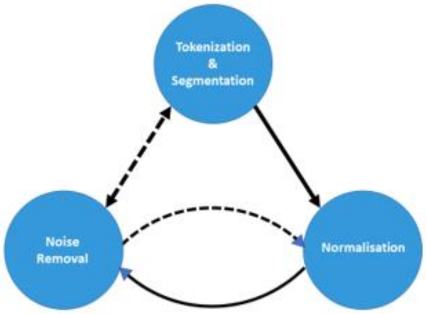

Fig. 4. Text data processing frame work

D. Sentimental Analysis in Information Extraction Systems

The most common text classification tool is Sentiment Analysis which analyses an incoming message and tells whether the related or implicit sentiment is positive, negative our neutral.

1) Subjectivity analysis

The subjectivity analysis has four components. In Subjective Sentence Classification The first component is a Naïve Bayes classifier that differentiate between subjective and objective sentences using a variety of lexical and contextual features Speech Act and Direct Subjective Expression Classification The second component identifies speech events (e.g., "said," "according to") and direct subjective expressions (e.g., "fears," "is happy"). Speech act include both speaking and writing events. Direct subjective expressions are words or phrases where an emotion, sentiment, etc. is directly described [14].

2) Part of speech tagging

Part of speech tagging (POS tagging) is one of the most important tasks of Natural Language Processing as it is responsible for attributing a label with grammatical information. This label is the part of speech of the word, which can be noun, verb, pronoun, preposition, adverb, conjunction, participle and article.

For example, the correct label can tell whether the meaning found in the text is correct, because words can have a different meaning when used as nouns and when used as verbs.

For example,

The word season: when used as a noun it means a period of the year (spring, summer, fall, winter), but when used as a verb means to apply spices or flavourings to food [15].

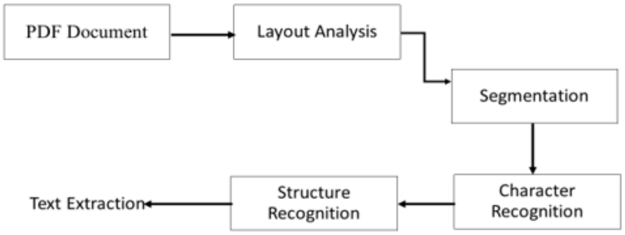

Fig. 5. Steps for text extraction

3) *Layout analysis*

The layout of a document here means the physical boundaries and locations of various regions in the document. As in Fig. 5 The sequence of Layout Analysis aims to break a document into a collection of homogenous regions, such as photos, background, text blocks, text lines, words, characters, etc. [16].

4) *Segmentation*

Segmentation algorithms are designed to handle complex document layouts and backgrounds, Document as shown in Fig. 6. Therefore, the target of the segmentation algorithm is to partitions a document with complex scripts [17].

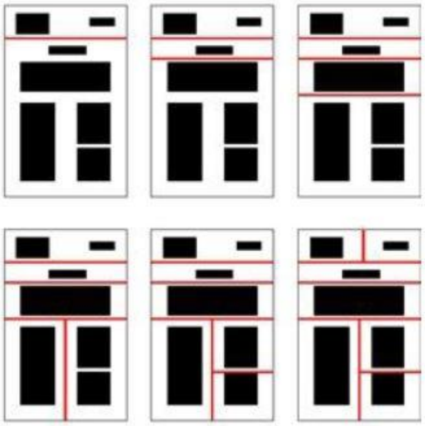

Fig. 6. An example for sequential cuts that segment a page into different regions

5) *Character recognition*

Optical character recognition is a system through which a system looks for a pattern in an image [18]. This conversion of image containing text to actual machine encoded text. PDF character recognition is the process by which characters are recognized from PDF files and placed into text searchable ones [19].

Applications:

1. Information extraction in digital libraries.
2. Information extraction from emails.
3. Person profile extraction.
4. Table extraction using conditional random field

4. Conclusion

This survey made an understandable and a simple overview on the finding of fact extraction from web content by going through various initial steps like information extraction and tokenization and many more consecutive processes. There are many working systems available in the market and also much improvisation is being done in research area. We expect the researchers to do advancements in each step of fact extraction that makes extracting the fact smooth in complex document format.

References

[1] Cunningham, Hamish. "Information extraction, automatic." Encyclopedia of language and linguistics, (2005): 665-677.
[2] Wasson, Mark D., et al. "System and method for extracting information from text using text annotation and fact extraction." U.S. Patent No. 7, 912,705. 22 Mar. 2011.
[3] Hassan, Naeemul, Chengkai Li, and Mark Tremayne. "Detecting check-worthy factual claims in presidential debates." Proceedings of the 24th ACM international on conference on information and knowledge management. ACM, 2015.
[4] Freitag, Dayne. "Machine Learning for Information Extraction in Informal Domains" (PDF). 2000 Kluwer Academic Publishers. Printed in The Netherlands
[5] https://www.nltk.org/book/ch07.html
[6] Sarawagi, Sunita. "Information extraction." Foundations and Trends® in Databases 1.3 (2008): 261-377.
[7] https://machinelearningmastery.com/supervised-and-unsupervised machine-learning-algorithms/
[8] Andersen, Peggy M., et al. "Automatic extraction of facts from press releases to generate news stories." Proceedings of the third conference on Applied natural language processing. Association for Computational Linguistics, 1992.
[9] Goddard, Cliff. Semantic analysis: A practical introduction. Oxford University Press, 2011.
[10] Fox, Chiristopher. "Chapter 7, lexical analysis and stoplists." Information Retrieval Data Structure and Algorithms ed. William B. Frakes, Ricardo Baeza-Yates Prentice Hall 1 (1992)
[11] Webster, Jonathan J., and Chunyu Kit. "Tokenization as the initial phase in NLP." COLING 1992 Volume 4: The 15th International Conference on Computational Linguistics. 1992.
[12] https://www.kdnuggets.com/2017/12/general-approach-preprocessing-textdata.html#:~:targetText=Stemming%20and%20lemmatization%20are%20major,of%20grammatical%20rules%20and%20norms.
[13] Méndez, José Ramon, et al. "Tokenising, stemming and stopword removal on anti-spam filtering domain." Conference of the Spanish Association for Artificial Intelligence. Springer, Berlin, Heidelberg, 2005.
[14] https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html
[15] Lewis, David D., and Karen Sparck Jones. "Natural language processing for information retrieval." Communications of the ACM 39.1 (1996): 92-101.
[16] Tamir Hassan, “Object-Level Document Analysis of PDF Files”, DocEng’09, September 16-18, 2009, Munich, Germany.
[17] Song Mao, Azriel Rosenfeld, and Tapas Kanungo 2003 Document structure analysis algorithms: A literature survey Vol. 5010 of SPIE Proceedings, SPIE, pp. 197-207.
[18] Williams S. Lovegrove and David F.Brailsford 1995 Document analysis of PDF files: methods, results and implications”, Electronic publishing, vol. 8 (2&3), 20-220.
[19] S. Audithan, R M. Chandrasekaran, “Document text extraction from document images using Haar Discrete Wavelet Transform”, EJSR, 2009.