TEST Engineering & Management

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

Survey of the Theory of Automata in Natural Language Processing

Aaron Carl T. Fernandez[1], Dr. Madhavi Devaraj[2], Ariel Kelly D. Balan[3]
[1],[2],[3]*Department of Computer Science, Mapua University, Manila, Philippines*
[1]*aaronfernadez1990@gmail.com*, [2]*mdevaraj@mapua.edu.ph*, [3]*akdbalan@mapua.edu.ph*

Article Info
Volume 83
Page Number: 5832 - 5839
Publication Issue:
May - June 2020

Article History
Article Received: *19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *17 May 2020*

Abstract

Words occur in sequence over time, and the words encountered constrain the meaning of the words that follow, rendering it critical in texts written or spoken in English, Spanish, etc. This makes finite-state machines effective at simulating the sequential property of a language. Having said that, this does not rule off other types of automata such as but not limited to, deterministic pushdown automaton and Turing machines, as these were shown to be useful in both transformational and generative grammars. The theory of automata has a rich literature in providing efficient and convenient tools for several branches of computational linguistics. This paper highlighted the significant works on that domain and presented some of the influential breakthroughs of the automata theory in the analysis of both syntax and semantics of the natural language.

Keywords: *Automata Theory; Natural Language Processing; Morphology; Phonology; Lexicon; Regular Grammars*

1. Introduction

Automata theory and natural language processing were once tightly bounded disciplines [1]. Russian mathematician Andrey Markov used finite-state processes to predict sequences of vowels and consonants in novels by Alexander Pushkin [2], emanating a technique now known as the Markov Chain. Claude Shannon continued this concept by using Markov processes to predict letter sequences of English words [3]. While most theorems about finite state machines were Automata theory and natural language processing were once tightly bounded disciplines [1]. Russian mathematician Andrey Markov used finite-state processes to predict sequences of vowels and consonants in novels by Alexander Pushkin [2], emanating a technique now known as the Markov Chain. Claude Shannon continued this concept by using Markov processes to predict letter sequences of English words [3]. While most theorems about finite state machines were proven in the 1950's, Noam Chomsky disputed that such devices were too simple to adequately interpret the natural language [4]. To deal with this, Chomsky suggested the use of context-free grammars and introduced the more powerful trans-formational grammars for the task at hand [5].

Having said that, the mainstream literature in automata theory and natural language processing eventually drifted apart. Automata theorists preferred theory-driven generalizations [6], [7] while linguists went the other way and abandoned formalism [7]. Although, some natural language processes still concentrated on context-free grammars extensions for a time [8], [9].

Eventually, speech recognition researchers returned to processing natural language grammar with finite-state machines by using transition weights that could be trained on machine-readable text datasets. These had algorithms that were efficient enough for practical computers back in the 1970's and were remarkably successful at determining correct from incorrect speech transcriptions [10], [11]. The interest in automata theory among computational linguists was brought back in the 21st century [12] – [14], specifically for problems like automatic language translation, wherein the transformations are sensitive to syntactic structure. It is without a

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

doubt that natural language processing is one of the major fields of application of this ideology [15].

The goal of this paper is to discuss the applications of automata theory in each level of natural language processing by surveying some of the significant literatures on the topic at hand. The history of natural language processing had begun way back in the 1950s, when the Georgetown experiment successfully developed a fully automatic translation of Russian sentences into English [1] but it wasn't until 2017 that Amazon's Jeff Bezos asserted that artificial intelligence, which strands include natural language processing, is entering its golden age, being able to solve problems that were just once seen as science fiction. An example of this feat is Amazon's voice assistant Alexa, which is an echo speaker that relies heavily on natural language processing to enable machines to understand human speech [16]. For this reason, the author of this paper confined this review to literatures published from 1956 until 2017, and had been indexed by INSPEC, Scopus, Web of Science (WoS), or DBLP.

2. LEXICONS:

Lexicons deal with the vocabulary of a person, group, or language in linguistics. It processes all the minimal grammatical elements of a language, which represents the speaker's knowledge of the vocabulary. It is comparable to a dictionary but without the definitions of the entries and instead, carries only part of words such as suffixes [15].

Fig. 1 shows the actual word and its grammatical information. The grammatical information can be its infinitive, past participle, or present form. It can also be its homonyms, adjective, noun or plural derivatives. These are useful when automatically processing a text since it allows automatic tagging of the text using only a simple text comparison. This naïve approach of representing a lexicon by simply enumerating the entry in a file is very expensive in terms of memory and computing usage [15].

```
did,V:PRET
do,N:s
do,V:INF
doe,N:s:p
does,V:P3s
does,N:p
done,V:PP
done,A
```

Fig. 1. A sample lexicon of the word "do".

An intuitive solution to optimize this is to sort the entries alphabetically to fasten the lookup but Revuz in his PhD thesis [17], showed that the efficient representation of lexicons is possible using some Boolean finite-state automata. His work proved that it is possible to attain a good compromise between the automaton size and speed of the access. Apart from the chosen representation for the automaton, determining it can significantly improve the access and minimizing it can reduce the number of states considerably. This algorithm becomes useful when the automata is handcrafted [17]. The process of building the automaton in this case is natural, and the determined and minimized automaton performs well in terms of processing time and memory consumption. Fig. 2 illustrates the automaton constructed for the lexicon Fig. 1.

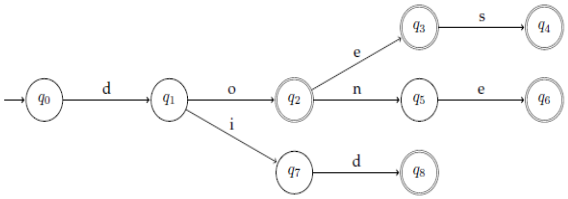

Fig. 2. The automaton equivalent of the lexicon in Fig. 1

Mohri in [15] reinforced this by replicating Revuz's thesis and by using lexicons in French, German and English. The study proved that using the more efficient finite-state machine implementation of lexicon lookup can improve the memory consumption by a factor of up to 18 [15].

3. MORPHOLOGY AND PHONOLOGY:

Morphology is the branch of linguistics that focuses on the study of the internal structure of a word and is considered as the smallest unit of syntax [18]. It analyzes the word production rules that are common to the speakers of a language. Phonology resembles morphology in the sense that it also studies the structure of a word, the difference is that it concentrates only on the spoken aspect or the sound patterns of the language. Both are based on the neologism phenomenon, which follows a wide range of rules such as the following:

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

Noun +" s" for the plural form (example: car, cars)

Verb + "s" for the present form (example: swim, swims)

Verb + "ed" for the past participle form (example: park, parked)

Verb + "er" to identify who (example: gamble, gambler)

Adjective + "ly" to qualify a way of doing things (example: gentle, gently)

Political figure + "ism" to refers to the ideology of a political figure (Example: Bush, bush-ism).

Some of the resulting words are not defined in most dictionaries but their meanings can be understood clearly as they are based on existing rules known innately by human [15].

This framework of the morphological ruling is quite similar to lexical semantics. Which means, its building process can also be intuitive, and its execution performance can be improved by the determinize and minimize algorithms.

Geyken and Hanneforth in [18] used finite-state automata to describe the morphological ruling for the German language. Named as "TAGH", the project achieved a 99% recognition rate based on 80,000 stem lexicon that was compiled within 5 years of German newspaper corpora and literary texts. It showed that the number of analyzable word forms in-crease considerably by more than 1,000 different rules for both compositional and derivational word formation.

Complete phonological rules could be easily defined using a finite-state machine as exemplified by the re-searches at the Xerox Alto Research Center [19]. The authors were able to simplify the evaluation of a grammar to the point of triviality using finite-state transducers. Without regards to the grammar complexity, the interpreter is said to be resistant to changes and the compiler becomes easier to implement [19].

Transducers are also useful to translation systems to represent their intermediate data as demonstrated by [20]. Their experiments consist of speech-input translations from Spanish to English and from Italian to English from telephone conversations between customers and front-desk of a hotel. The authors built a lexical automaton derived into a phonological representation as a pre-processing, which boasts of only 7.6% translation word error rate and 8.4% source-language speech decoding word error rate.

Unitex is another example of a program that embeds transducers representing word inflections. Fig. 3 is a sample automaton representation of word inflections in French, which can be combined with other lexical automaton to generate some part of the lexicon.

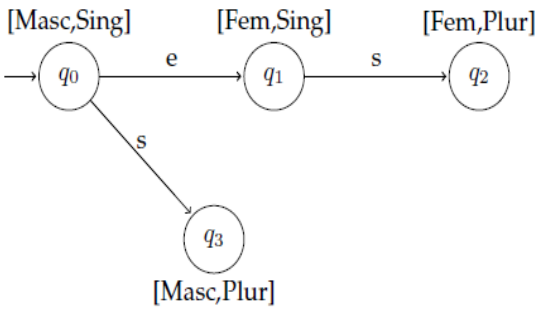

Fig. 3. Sample morphological inflection rule of French adjectives.

4. PART OF SPEECH TAGGING:

Part of speech tagging is the association of grammatical categories to words. It is usually performed first in a natural language process since most systems rely on its output to continue. Having said that, it usually suffers from various ambiguities, which is hardly resolved without the context of each word. The reason for this is that there is not just one correct tagging for a sentence, with each of the tags lead to different analyses of the sentence. An example would be the sentence "I read his book". The "I" in that sentence can either be a noun or a pronoun, both the "read" and "book" can be either be a noun or a verb, and the "his" can be an adjective, a pro-noun, or a noun. The problem arises when each word gives off a different meaning to the sentence depending on how it is used. A typical solution to this is to rely on the data provided by the lexicon but this could still be

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

problematic since the ambiguities are never solved and the system must handle these constantly.

Halteren, et al. in [21] used Hidden Markov Models to aid this problem. The main feat of their work is that it only required minimal resources and work while achieving 97.55% accuracy using only trigrams. Their work utilized three bench-mark datasets namely, the LOB corpus, which had proved to be a good testing ground for their task at hand, Wall Street Journal tagged with the Penn Treebank II tag set which is composed of roughly 1 million words and finally, the Eindhoven corpus tagged with the Wo-tan tag set, which is slightly smaller than the former, containing only around 750 thousand words. Although their results are positive, there are still a lot of research directions remain to be explored in their work. One of which is that better results can be obtained using the probability distributions generated by the component systems rather than just their best guesses. The other criticism is that there seems to be a disagreement on their paper be-tween a fixed set of component classifiers. But this pro-vides motivation for further extension of their work as there exist some dimensions of disagreement that may fruitfully searched to yield a larger ensemble of modular components that are evolved for a more optimal accuracy.

Shamsfard and Fadaee in [22] combined both probabilistic features and rule-based taggers to tag unknown Persian words. The distinction of this work is that their algorithm deals with the internal structure of the words and does not require any built-in knowledge. It is also domain independent since it uses morphological rules. To prove their work, the authors employed 300,000 words to calculate the morphological rules probabilities which were scraped from “Hamshahri” newspaper with a tag set containing 25 tags. Although this eliminates the bottleneck of the lexicon acquisition and the need to a preconstructed lexicon, the tradeoff for this is that it makes the tagger’s work more difficult. If some entries were entered in the lexicon then there would be probably fewer ambiguities in the tagging. It is also worth noting that although this work was done in Persian, the authors assured that it can also be applied to other languages as well.

Overall, the results of these probabilistic methods are good and comparable to the applications of the linguistic ruling that are exhibited by a linguist after months of analysis [21]. The amount is also comparable in terms of work. The linguist usually works on a dataset for months to write emphasis on the phenomenon [21], [22]. On the other side, the learning process can also be just as long on a real dataset, which can take up to months depending on the n-grams size considered [22]. N-grams are succession of N words commonly used with automatic systems to define the perspective of the learning process. It usually takes week when using bigrams and months when using trigrams [22]. Having said all of these, probabilistic system carries one major step back. The lack of readability in the systems produced hinders a more fruitful analysis. This is the main reason why most of the linguistic community prefer working on rules.

6. SYNTAX:

Syntax is the study of the rules that guide the correctness of a sentence in a language. Noam Chomsky implied a strong statement on his 1956 paper [4] “A properly formulated grammar should determine unambiguously the set of grammatical sentences”. This denotes that not only syntax should allow to decide if a sentence is grammatically correct, but it should also allow to define clearly if a sentence is semantically incorrect. However, this has not fully been attained yet as of this writing because of the irregularities which exists in the natural language [15]. There are several approaches that have been developed, most of which rely on lexicon. These are Lexical-Functional Grammars [23], Lexicalized Tree Adjoining Grammars [24], and Head-driven Phrase Structure Grammars [25]. These employ formalisms that associate grammar to the words, for instance, a verb learns whether it is transitive, etc. This paper will focus only on transformational and generative grammars since these are the most appropriate for automata application [4].

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

A. REGULAR GRAMMARS:

Regular grammars are defined by {Σ,V,P,S} where Σ is the alphabet of the language, V is a set of variables or non-terminals, P is a set of production or rewriting rules, and S is a symbol that represents the first rule to apply [26]. Fig. 4 presents an automaton for a regular grammar.

The rule for regular grammars is defined as follows, let α be a string of Σ^*, and let X, Y, and Z be a triplet of non-terminals. Any rules of P can be formed as follows [26]:

$$X \rightarrow Y\ Z$$
$$X \rightarrow \alpha Y$$
$$X \rightarrow Y$$
$$X \rightarrow \alpha$$

These have a small power of expression because of its restrictiveness. Thus, it can be utilized to handle small artificial languages or some restricted part of a natural language. It is also worth noting that these are totally equivalent to finite-state automata which makes it easy to use [26].

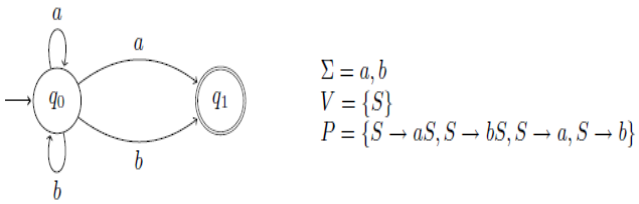

Fig. 4. A regular grammar and its corresponding automaton.

B. CONTEXT FREE GRAMMAR:

Context free grammars are like regular grammars, the only difference is that it does not have any restrictions on the length of the rules. These were first described by No-am Chomsky in [5]. What was interesting is that he developed this grammar to prove that it was insufficient for natural language processing. However, it became novel on the years that followed since it was the only implementable system that provided acceptable results. The expressiveness of these grammars remains acceptable if these are not used to describe the language with its whole complexity [5]. Fig. 5 presents a simple context free grammar that can perceive basic sentences like “the man loves the music, a firefighter risked his life, cat meows, or boy loves girls”.

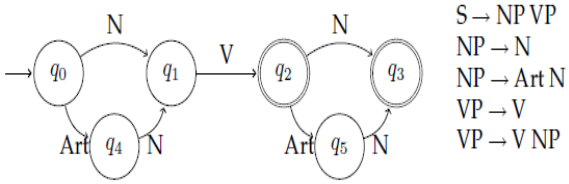

Fig. 5. A simple context free grammar and its corresponding automaton.

It is important to remember that context free grammars are not equivalent to finite-state automata [4]. Even though Fig. 6 insinuates that it may be represented by one, context free grammars are more synonymous to pushdown automata, but that will not be discussed on this paper as it does not hold any relevance. The reason for this is that context free grammars can generate languages like an bn which are not regular. An alternative is recursive transition network [27], which rely on the use of automata network and are equivalent to pushdown automata. This allows one automaton for each rule and evaluate this whenever a rule is reached on the transition. When the accepting state has been reached in the sub-automaton, it can go back to the previous automaton and resume treatment. This enables the automata in Fig. 5 to be translated into Fig. 6.

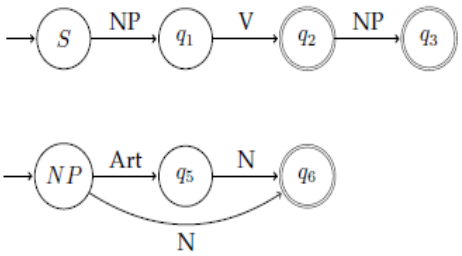

Fig. 6. Automata translation of Fig. 5.

To test this concept, an experimental parsing system was implemented in BBN LISP on the SDS 940 time-sharing systems at Harvard University and at Bolt by Beranek & Newman, Inx [27] and used for several of experiments in grammar development and parsing strategies for natural language analysis. The main criticism of this work though, is that the report of its results was not included in the paper as it was still in preparation as mentioned by the authors [27].

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

VI. INDEXING:

It is common to handle an enormous dataset of texts when working with natural languages. To work efficiently, it is imperative to have a fast access to the information required. To illustrate this, imagine the World Wide Web, which has billions of texts, but search engines can browse through it in only a few milliseconds to generate the queried information.

This can be achieved in natural language processing through a technique called indexing. The basic idea be-hind this is to create a database of text containing all the word occurrences and then instead of combing through the raw text data, the search is performed through the index assigned, which would be quicker.

Crochemore in [28] devised an algorithm using a finite-state transducer that allows a representation in a linear space of the text's index. The sum of the label weights during the recognition provides the first occurrence of a word in this algorithm, which then requires a specific path to retrieve all the other references. The author considered both the transducer and the automata to distinguish whether they deal with the suffixes or factors of the string. The size of the minimal automation is not the same in both cases in general. This is also to prevent the consideration of a marker at the end of a word.
This algorithm has been improved in [15], which stores the list of the current recognized word on each node. These word occurrences are then retrieved in linear time while maintaining the linearity of the storage space as well. Fig. 7 illustrates the automaton of the string "aabba". The length of the word must be subtract-ed to the list associated to that node to get the start of the current word.

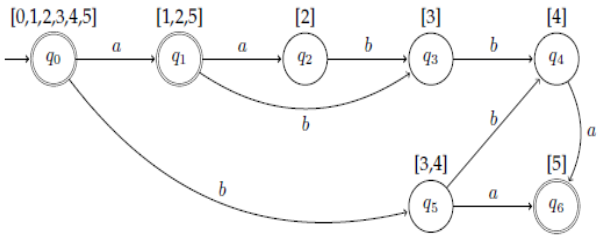

Fig.7. Indexing of the string "aabba" using an automaton.

VII. TRANSLATIONS:

Translation is the most complicated task in natural language processing as it suffers from the confluence of all the problems in linguistics [29]. To enumerate a few, one of these is the problem of lexical ambiguity in semantics. The word "bank" is a noun which has two distinct meanings: it may refer to a financial institution and it may also refer to the edge of a river. This is more complicated in translation because it combines the ambiguities of the two meanings at once.

This also poses a problem if part-of-speech tagging is to be applied in the translation process. The word "bank" can be either a verb or a noun, with each lead to different translation. Part of this problem stems from a syntactic point of view. It would be impossible to pro-duce the correct translation if a certain word has been perceived with the wrong category. Furthermore, even with the correct category, some of the syntactic structures are not well formalized yet. As an example, connecting a relative clause to its correct reference can become difficult when there are multiple relative clauses that are intertwined.

Problem in alignments also exists in the translation process. Most word and its translations vary in length tremendously. Take the word "potato" as an example, this word is translated to French as "pomme de terre". This brings up the problem of compound words and its detection. Although this can be solved using a lexicon, but some composition should be detected automatically because these are part of a regular construction, for instance, the use of "machine à + verb" in French.

This can be solved by using two lexicons, one in each language or by using hidden Markov models. Alshawi, et al in [29] proposed an algorithm based on a probabilistic transducer, which builds up a translation in three steps. First, it learns the alignment by using a bilingual dataset, after which, it creates the transducer using the same dataset and finally, the created transducer generates the translated version of the input. This has been developed by the authors to create an English to

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

Spanish translation model for a speech translation application, which achieved an accuracy of over 75% via string-distance comparison to three reference translations [29]. The training and test data for this experiment were taken from transcribed utterances from the Air Travel Information System (ATIS) corpus with a translated utterance in Spanish. An utterance is synonymous to a single sentence but can be more than one sometimes in a sequence [29].

This alignment algorithm has also been used in [30], the difference is that the authors based their experiment on N-grams translation models and variable n-gram stochastic automaton proposed in [31]. It uses a stochastic finite-state machine translation trained automatically from pairs of source and target utterances. This was developed for English-Japanese and Japanese-English translation. The main idea of this is to obtain the lexical translations through alignments algorithms and then to generate a variable n-gram stochastic automaton that will be transformed into a stochastic transducer which can reorder the output according to the language specifications. The data for their experiments were obtained from the customer side of an operator-customer conversation of a customer-care application. Each of the customer's utterance transcriptions were manually translated into Japanese. A total of 15,457 English-Japanese sentence pairs was split into 12,204 and 3,253 training and test sentence pairs, respectively. The main objective of their experiment is to measure the performance of their translation system in the context of an application. The number of sentences correctly translated on the number of expected sentence is 43.7, 62.5, and 65.5 on unigram, bi-gram, and trigram phrases respectively while the precision rate is 80.3, 86.3 and 85.5, respectively. Overall, the authors were successful at developing an architecture for speech translation in a limited domain based on the simple machinery of stochastic finite-state transducers. They have implemented a stochastic finite-state models for English to Japanese and Japanese to English translation, which have been trained automatically from source-target utterance pairs and evaluated in the context of a call-type classification task [30].

7. CONCLUSION

Natural language processing is a discipline which encompasses both computer science and linguistics. As automata theory is a significant component of theoretical computer science, it is only instinctive that it holds some use in processing natural language data. This paper bolstered that claim and exhibited how the automata theory can be applied in some of the stages in natural language processing. It discussed how it can efficiently represent both morphological and phonological rules, how useful regular and context free grammars are in confirming the correctness of a sentence in a language, how much memory and time it can save on indexation, and how it can solve the alignment problem during translation. There is a vast literature available on this topic, but the author of this paper chose only to highlight the important milestones in this field. A potential extension of this work is to explore its other facets, which have not been tackled on this paper, such as but not limited to automated speech recognition and information extraction. This is imperative to garner more appreciation and interest in the topic of the automata, as well as to epitomize its importance.

8. References

[1] Theoretical Computer Science Handbook of Weighted Automata, pp. 571–596, 2009.
[2] A. A. Markov, "An Example of Statistical Investigation of the Text Eugene Onegin Concerning the Connection of Samples in Chains," Science in Context, vol. 19, no. 04, p. 591, 2006.
[3] C. E. Shannon, "A mathematical theory of communication," ACM SIGMOBILE Mobile Computing and Communications Re-view, vol. 5, no. 1, p. 3, Jan. 2001.
[4] N. Chomsky, "Three models for the description of language," IEEE Transactions on Information Theory, vol. 2, no. 3, pp. 113–124, 1956.
[5] R. B. Lees and N. Chomsky, "Syntactic Structures," Language, vol. 33, no. 3, p. 375, 1957.
[6] S. Eilenberg, Automata, languages and machines. New York: Academic Press, 1974.
[7] F. Gécseg and M. Steinby, "Tree Languages," Handbook of For-mal Languages, pp. 1–68, 1997.

May – June 2020
ISSN: 0193-4120 Page No. 5832-5839

[8] M. Dalrymple, Lexical functional grammar. Leiden, The Netherlands: Brill, 2014.
[9] I. A. Sag and T. Wasow, "Syntactic Theory: A Formal Introduction," Computational Linguistics, vol. 26, no. 2, pp. 295–295, 2000.
[10] F. Jelinek, "Continuous Speech Recognition for Text Applications," Telecommunications Elektronische Textkommunikation / Electronic Text Communication, pp. 262–274, 1978.
[11] F. Jelinek, L. Bahl, and R. Mercer, "Design of a linguistic statistical decoder for the recognition of continuous speech," IEEE Transactions on Information Theory, vol. 21, no. 3, pp. 250–256, 1975.
[12] K. Knight and J. Graehl, "An Overview of Probabilistic Tree Transducers for Natural Language Processing," Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science, pp. 1–24, 2005.
[13] S. M. Shieber, "Synchronous grammars and transducers," Proceedings of the 10th International Conference on Parsing Technologies - IWPT 07, 2007.
[14] J. Graehl, K. Knight, and J. May, "Training Tree Transducers," Computational Linguistics, vol. 34, no. 3, pp. 391–427, 2008.
[15] M. Mohri, "On some applications of finite-state automata theory to natural language processing," Natural Language Engineering, vol. 2, no. 1, pp. 61–80, 1996.
[16] A. Kharpal, "A.I. is in a 'golden age' and solving problems that were once sci-fi, Amazon CEO Jeff Bezos says," CNBC, 08-May-2017. [Online]. Available: https://www.cnbc.com/2017/05/08/amazon-jeff-bezos-artificial-intelligence-ai-golden-age.html. [Accessed: 24-Sep-2018].
[17] D. Revuz, "Dictionnaires et lexiques: méthodes et algorithmes," Ph.D. thesis, Institut Blaise Pascal, Paris, France, 1991.
[18] A. Geyken and T. Hanneforth, "TAGH: A Complete Morphology for German Based on Weighted Finite State Automata," Lecture Notes in Computer Science Finite-State Methods and Natural Language Processing, pp. 55–66, 2006.
[19] R. M. Kaplan and M. Kay, "Regular models of phonological rule systems," Computational Linguistics - Special issue on computational phonology, vol. 20, no. 3, pp. 331–378, Sep. 1994.
[20] F. Casacuberta, E. Vidal, and J. M. Vilar, "Architectures for speech-to-speech translation using finite-state models," Proceedings of the ACL-02 workshop on Speech-to-speech translation: algorithms and systems -, 2002.
[21] H. V. Halteren, J. Zavrel, and W. Daelemans, "Improving Accuracy in Word Class Tagging through the Combination of Ma-chine Learning Systems," Computational Linguistics, vol. 27, no. 2, pp. 199–229, 2001.
[22] M. Shamsfard and H. Fadaee, "A Hybrid Morphology-Based POS Tagger for Persian,"Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), pp. 3453–3460, 2008.
[23] J. Bresnan, A. Asudeh, I. Toivonen, and S. Wechsler, Lexical-functional syntax. Chichester, West Sussex: Wiley-Blackwell, 2016.
[24] A. Sarkar and A. Joshi, "Coordination in Tree Adjoining Gram-mars," Proceedings of the 16th conference on Computational linguistics -, 1996.
[25] C. Pollard and I. A. Sag, Information-based syntax and semantics. Vol. 1: fundamentals. Stanford: CSLI, 1987.
[26] Martín-Vide Carlos, V. Mitrana, and Păun Gheorghe, Grammars and automata for string processing: from mathematics and computer science to biology, and back. London: Taylor & Fran-cis, 2003.
[27] W. A. Woods, "Transition network grammars for natural language analysis," Communications of the ACM, vol. 13, no. 10, pp. 591–606, Jan. 1970.
[28] M. Crochemore, "Transducers and repetitions," Theoretical Computer Science, vol. 45, pp. 63–86, 1986.
[29] H. Alshawi, S. Bangalore, and S. Douglas, "Automatic acquisition of hierarchical transduction models for machine translation," Proceedings of the 36th annual meeting on Association for Computational Linguistics -, 1998.
[30] S. Bangalore and G. Riccardi, "Stochastic finite-state models for spoken language machine translation," ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems on - EmbedMT 00, 2000.
[31] G. Riccardi, R. Pieraccini, and E. Bocchieri, "Stochastic automata for language modeling," Computer Speech & Language, vol. 10, no. 4, pp. 265–293, 1996.