Natural Language Processing Techniques for Enhancing Formation Evaluation

Klemens Katterbauer
EXPEC ARC
Saudi Aramco
Dhahran, Saudi Arabia
klemens.katterbauer@aramco.com

Rabeah Al Zaidy
Faculty of Computer Science
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
rabeah.alzaidy@kfupm.edu.sa

Sara Abu Al Saud
EXPEC ARC
Saudi Aramco
Dhahran, Saudi Arabia
Sara.abualsaud@aramco.com

Alberto Marsala
EXPEC ARC
Saudi Aramco
Dhahran, Saudi Arabia
Alberto.marsala@aramco.com

***Abstract*— Formation evaluation literature and reports in the oil and gas industry are crucial in decision making and understanding of optimizing recovery. The literature provides a comprehensive summary of tools and interpretations, as well as use cases for individuals to learn and utilize the information for enhancing their formation evaluation interpretations and decision-making. A major challenge in practice is the abundance and heterogeneity of information available that leads to individuals facing enormous obstacles to retrieving the right information within an adequate timeframe. We present an overview of several approaches in natural language processing for creating an ontology framework of formation evaluation data and literature, as well as conversational AI tools to extract information for the users. The review outlines the challenges that are faced when categorizing data related to formation evaluation, as well as establishing correlations and connections between various information sources. Finally, the review will provide a summary of different conversational AI approaches and systems for assisting well log and formation evaluation interpretation, as well as the opportunities and challenges faced. In conclusion, we will dedicate the way forward for NLP-driven approaches for assisting formation evaluation interpretation in real-time, and the business impact it has in the oil and gas industry and relationship to other initiatives both in the oil and gas industry as well as beyond.**

Keywords – reservoir formation evaluation; natural language processing; artificial intelligence; Petroleum industry

I. INTRODUCTION

Natural language processing (NLP) has become a cornerstone in several fields, including the oil and gas industry, and has become a cornerstone technology with crucial potential in the area of the 4th industrial revolution technology. NLP began in the 1950s as the intersection of artificial intelligence and linguistics. NLP was originally distinct from text information retrieval (IR), which employs highly scalable statistics-based techniques to index and search large volumes of text efficiently [1].

The statistical techniques utilized for IR encompass a wide range of frequency and distribution statistical methods. With time, however, NLP and IR have converged somewhat. Currently, NLP borrows from several very diverse fields, requiring today's NLP researchers and developers to broaden their mental knowledge base significantly. While statistical techniques represent a major area of NLP, advanced neural networks have become an important element to expand the utilization of NLP to learn in multiple settings by machines themselves. Simple statistical approaches face the challenge that it requires humans to provide and specify the dedicated responses for each human response. The response may differ depending on the context and in the light of the overall conversation, which made it almost impossible to compete with a human interpreter.

Word-for-word Russian-to-English machine translations, due to their primitive nature, were in the early days easily defeated by homographs[1] and metaphor. For example, the statement "the spirit is will, but the flesh is weak," was translated into "vodka can be agreed on, but it spoiled the meat," which easily showed the limitations and potential wrong conclusions that may be derived from word-for-word translations [2]. While the test failed tremendously, it provided a breakthrough for the computing industry, which showed that a computer is able to provide machine translations.

The first theoretical analysis of the complexity of language grammar was carried out by Chomsky [3]. This significantly influenced the creation of the Backus-Naur Form (BNF) notation, which is still widely utilized [4]. The focus of BNF is to define context-free grammar in a similar form as a programming language syntax. The main objective is to translate context-free grammar into a form that is

[1] Identically spelled words with multiple meanings

understandable for computer scientists and can be easily implemented on a computer.

When analyzing a language, the BNF specification consists of a number of derivation rules that syntactically validate the program code. A crucial understanding in this context is that the rules do not represent expert systems heuristics but solely constraints.

Another crucial part was the development of text-search patterns, based on which the concept of regular expression syntax was developed [5].

These developments led in the 1970s to heavily exploit lexical-analyzer (lexer) generators and parsers that incorporated grammars. A lexer is a transformer that transforms a text into tokens, where the subsequent parser validates the sequence of the tokens. The combination of lexers and parsers provides a solid foundation for the implementation in a programming language as it takes regular expressions and the BNF specifications and transforms it into code and lookup tables to determine decisions related to lexing and parsing [6].

Although context-free grammar (CFG) may not theoretically be adequate for natural language processing, its ability to transform easily into programming language syntax makes them very attractive in practice [7]. This has to do with the fact that there is a deliberate attempt to have a restrictive CFG variant in order to improve the implementation. Such a form of grammar is called a look-ahead parser with left-to-right processing and rightmost (bottom-up) derivation (LALR) [8]. The operating procedure of LALR is that the text is scanned first of all from left to right and then performs a bottom-up approach, where the compounds are constructed gradually from simpler ones. The look-ahead implies that the parsing decisions are made based on taking into account a single token ahead of the existing token. Given that there is only a single token that is taken into account when determining a parsing decision, this may represent a challenge to adequately infer the meaning of a sentence structure [9].

The 1970s also led to the development of the Prolog language, whose syntax is focused on writing grammars [10]. In order to achieve the simplest implementation mode (top-down parsing), the rules have to be changed to right-recursively. The challenge with a top-down approach is that they are considerably slower than bottom-up parsers, as they do not need generators.

II. Statistical NLP: Overcoming the challenges of specified, explicit rules

The difference between various natural languages differs tremendously, which exacerbates the challenge of determining the intent and meaning of sentences and statements in a specific language. The huge size, as well as unrestricted nature of natural languages, present significant problems that are further exacerbated by the ambiguity of language [11]. Hence, standard parsing approaches based on symbolic and manual rules are set to face two major critical challenges (Figure 1).

- The first challenge is that NLP has to extract the meanings of the text, which are the semantics. These are the formal grammars that outline the relationship between the text units. [2] that address primarily syntax. Extension of grammars by expanding sub-categorization and incorporating additional constraints and rules can help understanding better the natural language semantics; however, the increasing number of rules can lead to an unmanageable set that may unpredictably interact with each other and can lead to multiple interpretations of the word sequence. The arising ambiguity represents a major challenge, as the user is interested in the context and avoids ambiguity in interpretation.
- Another challenge is that handwritten rules face significant challenges with ungrammatical spoken sentences, even though the sentence is comprehensible by humans.

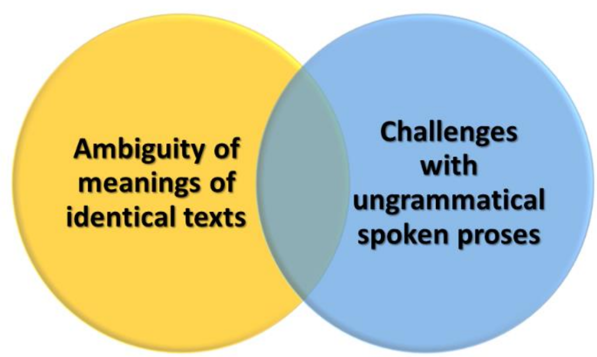

Figure 1: Major challenges of handwritten rules-based NLP.

These two challenges led to a significant rethinking of how to approach the processing of natural language via focusing on simple and robust approximations of the natural language instead of deep analysis (Figure 2). Additionally, evaluation became considerably more rigorous as compared to before, and the utilization of machine learning techniques.

The move from deterministic to probabilistic language models was a decisive factor given the inherent ambiguity of language and also the probabilistic determination of the meaning of sentences by humans themselves. Almost anyone has experienced that the meaning of a sentence or prose may very much differ in the context or how it is spoken. The same form and way how a prose is stated may even differ in terms of its interpretation between different cultures [12].

[2] These are parts of speech, such as nouns, verbs, and adjectives

Additionally, larger documented text statements were utilized for training these new machine learning algorithms, which provided a ground truth for the evaluation, and hence better determination of how to correctly interpret the text fragments and sentences.

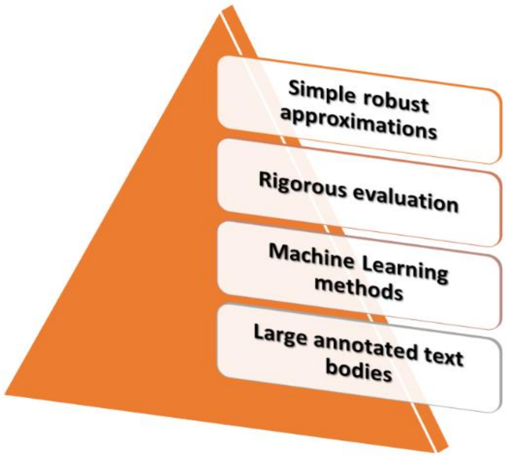

Figure 2: NLP reorientation in the 1980s.

This reorientation led to the rise of statistical NLP where statistical parsing utilizes probability for the context-free grammar rules [13]. Each rule has an associated probability, which is typically derived via machine learning on a described text corpora. This is also considered to be a supervised machine learning approach that represents an important part in NLP. The advantage of such an approach is that very detailed rules are replaces with statistical-frequency information lookup to avoid the ambiguity that may arise.

A different approach is that the rules are created from the annotated data, which builds then a decision tree from the feature-vector data. The statistical parser evaluates the highest probability for a parse of a phrase or sentence and then utilizes this parse to process the sentence and assign a meaning. The probabilistic approach depends considerably on the context, however, so having an acceptable training corpus is essential [14]. A training set consisting of annotated text bodies from the Wall Street Journal may be unsuitable for formation evaluation, as many words and meanings are not incorporated into the training set.

The main advantage of statistical approaches in practice is that the algorithms train with real data and utilize the most common cases. This implies that the more abundant and representative the data are for the phrase or text under consideration, the better they get. Another advantage is that unfamiliar or erroneous input may lead to lesser challenges, given that they indicate a low probability of matching. Handwritten rule-based and statistical approaches are complementary with each other, which is crucial for the success of NLP approaches.

III. Application of NLP in Reservoir Formation Evaluation

Within NLP there are typically several sub-problems that can be gradually addressed and solved, such as speech synthetics and connected speech recognition. Question answering, especially in technical domains, represents a major challenge.

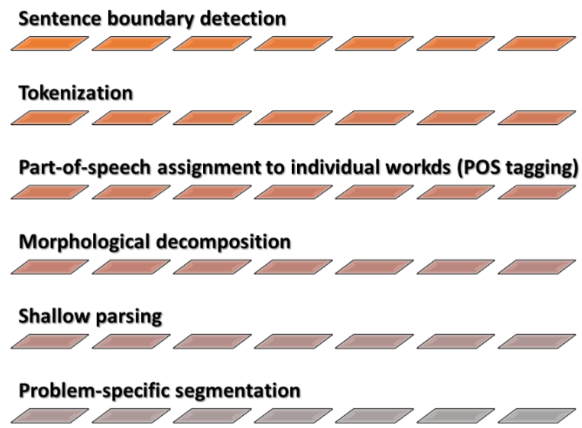

Figure 3: Low-level NLP tasks for reservoir formation evaluation.

Conventional low-level NLP (Figure 3) tasks involve sentence boundary detection, where the end of a sentence is to be looked for [15]. Conventionally, this is rather simple, given that a full stop ends a sentence. Abbreviations and titles represent a considerable challenging task in addition to items in a list.

Another task is tokenization, which identifies the individual tokens in a sentence. These tasks are conventionally covered by lexers. However, characters, such as dashes and forward slashes, may cause issues as they do not necessarily separate different tokens.

Part of speech tagging represents another challenge as they may represent a verb as well as a noun in certain circumstances. This involves the use of -ing that may be used in both verbs and nouns [16].

Another challenge is morphological decompositions of compound words that require a decomposition of the word to comprehend them. This is especially true for technical disciplines that contain many technical terms, which are hard to understand by themselves. Lemmatization typically helps in this context, but this depends on the language under consideration [17].

Shallow parsing is another low-level NLP task that identifies phrases from tokens that are tagged as part of the speech. For example, an adjective may precede a noun, which it describes [18].

Segmentation, according to specific problems, represents another challenge that is rather low-level.

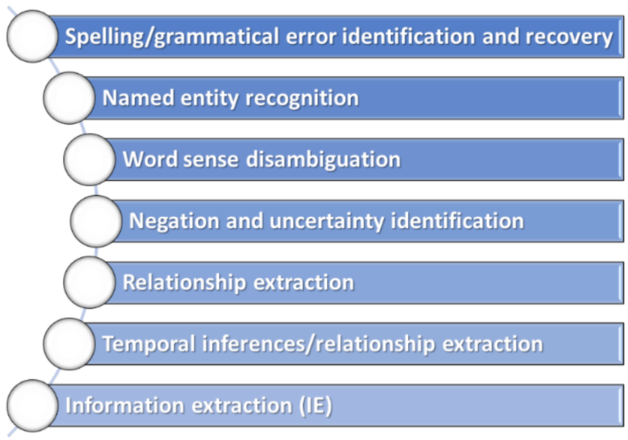

Figure 4: High-level tasks for NLP in reservoir formation evaluation.

Higher-level tasks build on low-level tasks and are usually problem-specific (Figure 4). They include:

1) **Spelling/grammatical error identification and recovery:** This is a very interactive task but is not that perfect from an implementation perspective. These phrases may lead to false positives, which are words that are correct but are flagged as false. Homophones may be used incorrectly and lead to false negatives. Typically, a homophone for reservoir formation evaluation is “their” and “there” [12].
2) **Named entity recognition (NER):** NER stands for the identification of entities, which are specific words or phrases, and then categorizes them into entities, such as persons, machines, locations, etc. The most common task is to develop a mapping between the named entities and concepts in a vocabulary, which partially utilizes shallow parsing. This may be separated into multiple phrases, however [19].
3) Some major issues that are faced in NER are:
   - Word/phrase order variation: This may be, for example, formation reservoir evaluation in contrast to reservoir formation evaluation
   - Derivation: This may lead to the derivation of suffixes
   - Inflection: This may be, for example, changes in numbers
   - Synonymy is abundant in formation evaluation and engineering.
   - Homographs**:** Homographs with related meanings are called “polysemy” and there are numerous examples of such.
4) Word sense disambiguation (WSD): This involves the determination of the correct meaning.
5) Negation and uncertainty identification: uncertainty identification has become essential, as synonyms or named entities are widespread encountered. Determining the absence or presence as well as quantifying the inference's uncertainty is a major challenge. Negations, on the other hand, can be explicit but can also be expressed in the form of uncertainty, which allows one to hedge. When talking about uncertainty, one determines that the reasoning process is hard to understand.
6) Relationship extraction: A crucial part is to determine relationships between an entity and events that are taking place. This is commonly encountered in formation evaluation and referencing to a thesauri or databases typically assists in overcoming this challenge and helps to extract the relationships.

Another sub-task for determining relationships between entities that are hierarchically related is called anaphora reference resolution [20]. This includes:

- Identity: In formation evaluation there are many instances where there are pronouns that refer to a named entity or where an abbreviation is used after the first time mentioning.
- Part/whole: This occurs when there is a location within a field;
- Superset/subset: For example, formation evaluation, logging.

7) Temporal inferences/relationship extraction: This refers to the inference from expressions or relations that are temporal. In particular, studying the past may allow inferring whether an event may occur in the future again or order the narrative.
8) Information extraction (IE): This refers to the identification of information that is problem-specific or focuses on the transformation into a structured form [21].

IV. Statistical Machine Learning – Data-Driven Approaches

Statistical and machine learning is a well-known area that involves the development of algorithms for inferring patterns from data. This shall help to be able to generalize and make predictions for new data via learning from the previously recorded data [22]. The process is typically separated into the training and prediction phase, where the parameters of the algorithm are optimized in order to minimize the discrepancy between the expected numerical target and the estimated.
The learning can be either supervised or unsupervised. In the supervised instance, the items in the training data are correctly labeled. In the unsupervised instance, the training data are not labeled, and the training process tries to determine the pattern automatically. This may be in the form of a cluster or factory analysis, or various other approaches [23].
One of the major challenges faced for any learning approach is overfitting. This implies that the model fits the data almost perfectly; however, the predictions for new data are rather poor. This is a major challenge if the data are not

representative of the instances one will face in the future or if the model is very erratically behaving [12].
This is primarily due to the fact that the models learn the random noise in the training data instead of retrieving the essential features that are desired. A great way to overcome the challenge of overfitting is to utilize cross-validation, which partitions the training dataset into tests and training sets where these are then internally validated. When repeating this process over several rounds, wherein each step the data are partitioned randomly, it allows to obtain a better average of the performance of the model and improve it [22].
Machine learning can be further classified according to how the probability distributions are utilized. Generative methods have the aim to create probability distributions for models, which allows the model to create synthetic data with these probability distributions. A more utilitarian approach is to use discriminative methods that estimate based on the observations directly the posterior probability.
In natural language processing, a generative approach would be to utilize in-depth knowledge of various languages to determine the undetermined language of a speaker, while a discriminative approach would utilize the difference between the various languages and the spoken language and then try to find the closest match.
The challenge of generative models is that they relatively easily become intractable for more features. In contrast, discriminative models have the benefit that they allow more features.
Typical examples of discriminative methods are logistic regression and conditional random fields (CRFs), while generative methods encompass Naïve Bayes classifiers and hidden Markov models (HMMs) [22].
There are, however, several major machine learning methods that are most often used for natural language processing tasks in formation evaluation [13].

Support vector machines (SVMs)

Learning via a discriminative approach is achieved via support vector machines (SVMs) that utilize inputs, such as words, to classify them into categories. This may be part of speech or other classification forms. The input in the SVM is conventionally transformed in order to enable the linear separation of the data into various categories. A crucial part of this is the transformation function, also called the kernel function, that transforms the data [24].
To outline the application of support vector machines, in a two-feature case, such as classifying a written report in terms of whether it categorizes a productive formation or nonproductive, typically can be separated by a straight line if solely two input features are utilized (see Figure 5). For the case of N-features, the separator will be conventionally an N-1 hyperplane, where the separating hyperplane aims to maximize the distance between the support vectors for each category. The support vectors are the data points that are closest to the hyperplane that differentiates each category. The most widely utilized kernel function for the transformation utilizes the normal distribution, given that in lots instances, the data are normally distributed [25].

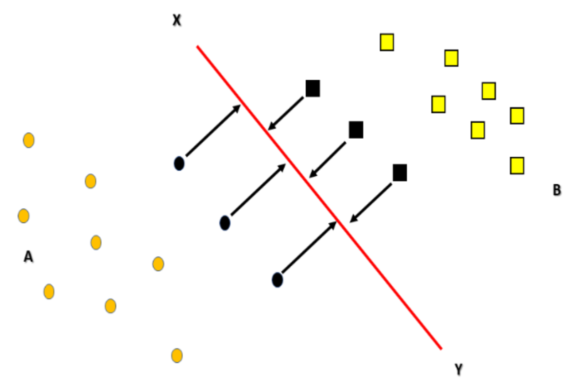

Figure 5: Support vector machines: We outline a 2-D case, where the points are separated by a straight line. The data are categorized in two categories, specifically category A (circles), and category B (diamonds). The data points can be separated by a straight line in the 2-dimensional plot. The SVM algorithm identifies the points that are closest to different categories and then determines the line that maximizes the margin between both sides. Linear separation may not always be feasible. Hence a transformation via a kernel function is necessary. This requires, in many instances, a trial and error approach in case the distribution of the data or transformation to allow linear separation is unknown.

Hidden Markov models (HMMs)

Hidden Markov models are systems that allow variables to move between different states that leads to various output possibilities. The move between the various states depends on the probabilities of the moves, which then also encounters various probabilities. The word "hidden" in HMM refers to that the system's state-switch probabilities and output probabilities being hidden, while only the outputs are known. While the number of possible states and unique identifiers may be large, they are still finite and known (see Figure 6) [26]. There are several crucial aspects in hidden Markov models.

- **Inference:** Inference refers to the computation of the probabilities of one or multiple candidates for a state-switch sequence.
- **Pattern matching:** Pattern matching refers to the switch sequence between the states that are with a high probability generating the output-symbol sequence.
- **Training:** Whenever the output-symbol sequence data are known, then the state-switch/output probabilities can be computed in terms of that it best fits the data.

The pattern matching and the training are similar to Naïve Bayesian reasoning extended to sequences, which can then be considered a generative model [27]. The main simplifying assumptions utilized for HMM are that

1) the state switching probability depends strongly on the states previously. This may also allow to switch back to the same state. In the simplest case, where there is only one state, the current state alone determines the probability. Hence, HMMs of the first order are very useful for situations where the likelihood solely depends on the last event and not the previous.
2) A specific output has a probability that solely depends on the state and no other state.

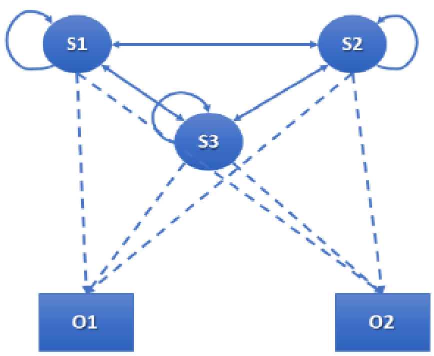

Figure 6: A graphical illustration of hidden Markov models. The rectangles with the letter O refer to the output values, whereas the circles starting with the letter S represent the states. The solid lines represent the state switches between connected states, where the arrow allows to indicate the switch's direction. It is noteworthy that the states may switch back to themselves with a certain probability. The probability may differ for the various lines. The dashed lines connect the states to the output values, which allows inferring the output probability. Important to note is that the sum of the probabilities of a switch leaving it is equal to 1, as this ensures consistency that all possible state transitions are considered.

The underlying assumptions enable to easily calculate the probability of a state switch sequence via simple multiplication, which can be easily addressed with algorithms such as the Viterbi algorithm. There are various problems in reservoir formation evaluation, in particular when considering the sensing part, that can be addressed with these existing algorithms [28].

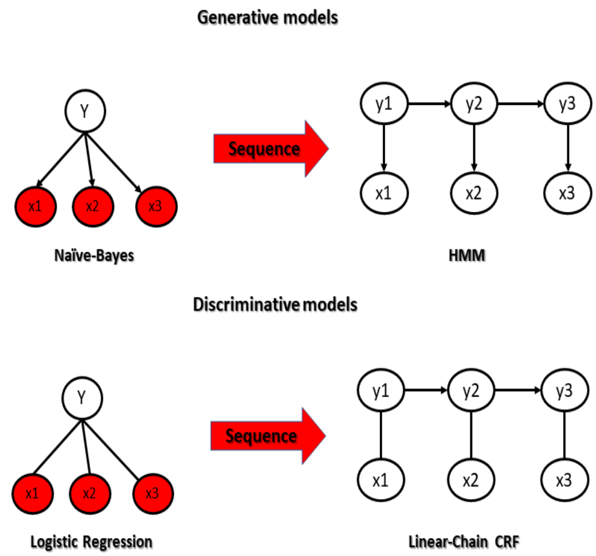

Figure 7: We outline the relationship between the Naïve Bayes, logistic regression, and conditional random fields. Naïve Bayes and Logistic regression distinguish each other from that Naïve Bayes is a generative model, while logistic regression is a generative model, which can be either transformed for sequences in a hidden Markov model or into a linear conditional random fields model. The dependence is indicated in both instances by the directional arrows that show the dependence between the various states.

The extension of HMMs to multivariate scenarios is possible. However the challenge arises from the potential intractability of the training problem. This leads to that multiple-variable applications deploy single variables, partially artificial, in order to determine the composites of the categorical variables. This requires much more training data to be available. When referring to speech recognition, the word's waveform, in terms of how it is spoken, is then connected to a sequence of the individual states (phonemes) that may be best at reproducing it. While speech recognition has improved significantly, formation evaluation still faces challenges in the field due to the complex terminologies and similarities between words.

Conditional random fields (CRFs)

Conditional random fields are discriminative forms, where the linear chain form of CRFs resemble hidden Markov models in that the next state solely depends on the current state. This indicates a linear dependency, which allows for fast and efficient computation. The conditional random fields are primarily a generalization of logistic regression to sequential data as compared to the previous discussion of the extension of Naïve Bayes to HMM (see Figure 7).

CRFs are widely applied to NER challenges, where the state variables are the categories of the named entities. Then the objective is to predict the sequence of named entity categories within the phrase or word pattern. The observation may involve prefixes and suffixes, as well as capitalization and embedded numbers. Hyphenation may be applied. For formation evaluation aspects, a well needs to be succeeded by an entity that has to be a number. If the field is called “Resfield,” then to indicate a specific well, the named entity needs to be followed by a number, such that it states, for example, “Resfield 1.” The main benefit from CRF is that it can be easier applied to sequential multivariate data as compared HMMs, as the training problem will be tractable.

N-grams

N-grams are powerful tools in statistical machine learning, where an n-gram is a sequence of n items that may consist of letters, words, or phonemes. Certain item pairs may occur with various statistical frequencies, where the relationship between various characters may be easily determined. This connection depends on the language under consideration, as certain combinations of word characters are rather unusual. The challenge in reservoir formation evaluation is that there are lots of abbreviations which makes the distribution broader. However, if sufficient data are available, then the frequency distribution for the n-grams can be computed. The permutations may increase dramatically, as in English alone, there are 26^2 letter pairs alone, which n-tuplets amounting to 26^n possible forms. This shows that n-grams depend on the n-th position on the previous n-1 items that were computed from the data.
The n-gram data has several purposes:

- Auto-completion suggestions of words, phrases, wells, etc. that are widely encountered on smartphones.
- Correction of misspelled words or names can be done automatically. This may also refer to reservoir names.
- For speech recognition, the ambiguity can be reduced based on determining the neighboring words
- probabilistically.
- The word "well" may have different meanings. In formation evaluation, it primarily relates to a noun, while in normal English, it is typically referred to as an adverb. Given the non-ambiguous neighboring words, the correct meaning of the homograph can be easily determined.

The challenge with n-grams is that they are voluminous, and this may become a challenge when retrieving the data. With modern data structures, such as n-gram indexes, searching of such data can be significantly sped up. The advantage is that n-grams need relatively little linguistic and domain knowledge.

V. NPL ANALYTICAL PIPELINES

For any NLP tasks, there are typically several sub-tasks that need to be focused on. These sub-problems require these low-level tasks to be executed sequentially before any higher-level task can be started. Hence, a pipelined design system is crucial as the output of one module may be connected to another module, which allows for mixing and matching of the various approaches.
For example, a CRF may be combined with a named entity recognition framework, which improves robustness. A single module could be easily replaced with another without having to conduct substantial changes to the remainder of the system [29].
A famous pipelined NLP framework is the Unstructured Information Management Architecture, where the scope allows to integrate structured-format databases, images, and multi-media, in addition to arbitrary technology.
This becomes even more important for reservoir formation evaluation applications that require a multi-step pipelined approach to move from voice interpretation over to technical specification understanding, over to automatic interpretation and recommender engines.

VI. THE FUTURE OF NLP IN FORMATION EVALUATION

Recent advances in artificial intelligence have outlined the importance of NLP in formation evaluation, and the huge potential encountered in the area. The large disk capacities, as well as data compression and efficient search allows modern statistical NLP methods to mimic human thoughts and speech patterns.
Multipurpose NLP technology will become mainstream for well log interpretation, and well report summary creation, which will also incorporate the automatic analysis of drilling reports for crucial information.

VII. CONCLUSIONS

Natural Language processing has in the last century undergone a revolution from being a fringe technology to powering many tools and services in today’s environment. Formation evaluation represents a crucial area where natural language processing can play a vital role for enhancing interpretation and subsurface understanding. This will go beyond just pure textual understanding over to automatic speech recognition and interpretation, as well as hands-free tool deployment and automation.

VIII. REFERENCES

[1] K. Katterbauer, A. Marsala, R. Alyami, R. Al Zaidy, “An overview of natural language processing driven approaches towards assisted formation evaluation interpretation“, in Proc. *ICSNC, 2020.*

[2] C. Manning, P. Raghavan and H. Schuetze, Introduction to Information Retrieval, Stanford: Cambridge University Press, 2008.

[3] J. Hutchins, "The first public demonstration of machine translation: the Georgetown-IBM system," 7th January 1954.

[4] N. Chomsky, "Three models for the description of language," *IRE Transactions on Information Theory,* vol. 2, no. 3, pp. 113-124, 1956.

[5] R. Heckendorn, "A practical tutorial on conext free grammars," University of Idaho, 2020.

[6] J. Friedl, Mastering Regular Expressions, New York: O'Reilly Media, 2006.

[7] M. Zahran, "Lexical Analysis I," New Yrok University, New York, 2012.

[8] H. Bordihn, "Mildly Context-Sensitive Grammars," *Formal Languages and Applications,* pp. 163-173, 2004.

[9] J. Levine, Flex & Bison: Text Processing Tools, O'Reilly Media, 2009.

[10] S.-i. Morimoto and M. Sassa, "Yet another generation of LALR parsers for regular right part grammars," *Acta Informatica,* vol. 37, no. 9, pp. 671-697, 2001.

[11] M. Bramer, Logic Programming with Prolog, Portsmouth: Springer, 2005.

[12] D. Klein, "A core-tools statistical NLP course," in *Proceedings of the Second ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL*, 2005.

[13] W. Heng, "Natural Language Processing - Introduction and Practice," Amazon China, Beijing, 2013.

[14] P. Kantor, "Foundations of statistical natural language processing," *Information Retrieval,* vol. 4, no. 1, p. 80, 2001.

[15] B. Larsen, "A trainable summarizer with knowledge acquired from robust NLP techniques," *Advances in automatic text summarization,* p. 71, 1999.

[16] M. Straka and J. Strakova, " Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 2017.

[17] A. Voutilainen, "Part-of-speech tagging," *The Oxford handbook of computational linguistics,* pp. 219-232, 2003.

[18] E. Roche and Y. Schabes, Finite-state language processing., MIT Press, 1997.

[19] J. Hammerton, M. Osborne, S. Armstrong and W. Daelemans, "Introduction to Special Issue on Machine Learning Approaches to Shallow Parsing," *Journal of Machine Learning Research,* vol. 2, no. 4, 2002.

[20] A. Mansouri, L. S. Affendey and A. Mamat, "Named entity recognition approaches," *International Journal of Computer Science and Network Security 8,* vol. 2, pp. 339-344, 2008.

[21] L. Pineda and G. Garza, "A model for multimodal reference resolution," *Computational Linguistics,* vol. 26, no. 2, pp. 139-193, 2000.

[22] C. Cardie, "Empirical methods in information extraction," *AI magazine,* vol. 18, no. 4, p. 65, 1997.

[23] S. Masashi, Introduction to statistical machine learning., Morgan Kaufmann, 2015.

[24] J. Lafferty and L. Wasserman, "Challenges in statistical machine learning," *Statistica Sinica,* vol. 16, no. 2, p. 307, 2006.

[25] W. Noble, "What is a support vector machine?," *Nature biotechnology,* vol. 24, no. 12, pp. 1565-1567, 2006.

[26] L. Wang, Support vector machines: theory and applications, Springer Science & Business Media, 2005.

[27] O. Cappe, E. Moulines and T. Ryden, Inference in hidden Markov models, Springer Science & Business Media, 2006.

[28] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[29] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," *Hidden Markov models: applications in computer vision,* pp. 9-41, 2001.