Challenges for Information Extraction in the Oil and Gas Domain*

Alexandre Rademaker[1]

[1]IBM Research and FGV/EMAp
Rio de Janeiro, Brazil

alexrad@br.ibm.com

Abstract. *Increasingly, governments, corporations, and scientific organizations need to extract complex information from highly technical documents. While linguistic resources exist in some technical domains, they are largely unavailable for the oil and gas domain. We applied natural language processing tools with minimum domain adaptation to extract information from 155 annotated text passages from geological reports. In recognizing oil field entity names, we achieved a precision of .94 and recall of .43 (F1=.59) without supervised learning. We describe the impact of errors found in the output, including incorrect segmentation, part-of-speech tags, multiword expressions, word sense disambiguation, numeric quantities, and other issues leading to incorrect entity classifications. These mistakes could be reduced with a domain-specific dictionary that includes part-of-speech tags.*

Resumo. *Cada vez mais governos, corporações e instituições científicas precisam extrair informações complexas de documentos técnicos. Enquanto recursos linguísticos existem em alguns domínios técnicos, estes estão em grande parte indisponível para o domínio de petróleo e gás. Nós aplicamos ferramentas de processamento de texto com mínima adaptação ao domínio para extrair informações de 155 passagens de texto de relatórios geológicos anotados. Ao reconhecer os nomes das entidades dos campos de petróleo, alcançamos uma precisão de .94 e um recall de .43 (F1 = .59) sem aprendizagem supervisionada. Nós descrevemos o impacto dos erros de segmentação de sentenças, tagging, identificação de expressões multi-palavra, desambiguação do sentido das palavras, e outras questões, na classificação incorreta das entidades. Os erros encontrados poderiam, em sua maioria, serem evitados com um dicionário específico de domínio.*

1. Introduction

Oil exploration and production companies annually invest billions of dollars gathering and processing data contained within documents, including reports, scientific articles, business intelligence articles etc. These documents include critical information that drives important decisions such as whether to drill exploratory wells, bid or buy, and production schedules. Additionally, this unstructured data is growing exponentially each year and

*The author would like to thank Fabricio Chalub, Shari Trewin, Robert Farrell for contributing to a previous version of this material presented here. Henrique Muniz helped with some experiments.

organizations are finding the management of unstructured data to be one of their most critical challenges [Antoniak et al. 2016, Palkowsky 2005, Feblowitz 2013].

Information extraction from unstructured text requires a sequence of natural language processing steps, a linguistic pipeline, including the text segmentation in tokens and sentences, definition of the entities of interest, detection of mentions of these entities in the text (mention detection), linking mentions that refer to the same thing in the world (co-reference resolution), and extracting relations between the detected entities (relation extraction). There has been considerable work on mention detection, particularly detection of named entities [Ratinov and Roth 2009], including development of algorithms in specific technical domains, such as biomedical, finance, and chemistry. In oil and gas (O&G), named entities include named oil fields, basins, rock formations, and so on. Basic techniques include regular expressions and similarity based approaches. Regular expressions can take advantage of regularities in language often introduced in technical domains. Similarity-based techniques, such as Statistical Character-based Syntax Similarity (SCSS), can handle character-level, word-level and word order variations when matching text to dictionary entries [Tohidi et al. 2014]. However, adapting these algorithms to a new domain requires significant work. Thus, in recent years, researchers have turned to supervised machine learning methods.

With supervised learning approaches, people annotate a set of documents (the training data) manually, often using an annotation tool with a user interface, providing labels for text segments, used to train a statistical model. Afterwards, the model can be used to annotate similar documents, assigning the correct labels to new text segments. There are many weaknesses in this approach. The most important limitation is that the training data must be similar to the data that will be annotated with the model. Text annotation is also very time-consuming, and in specialized domains it requires domain expertise, making it impossible to use less expensive crowdsourcing approaches.

The cost of developing a model for a new domain, such as O&G, can be mitigated by reusing annotations from a different domain, but also correcting those annotations using rule-based transformations informed by known differences in the lexicon and linguistic constructs special to the new domain.

This article describes an approach to information extraction that is informed by linguistics. We developed a workbench of tools to understand the shortcomings of a current statistical entity and relation extraction system. Our goal is to achieve competitive results without the significant cost of creating an annotated training corpus.

The shortcomings of current NLP methods on the highly complex, domain-specific language in the O&G domain are not well described. This article elucidates some of the reasons why statistical methods, while generally outperforming rule-based methods, still have challenges in this difficult domain. We argue that with the right investments in the construction of lexical resources and corpora, we can improve results and the ability to reuse, adapt and replicate resources from other domains.

2. The statistical-based approach

Regarding supervised learning approaches for named entities or mention extraction, there are many tools available. Tools such as GATE [Cunningham et al. 2011] or IBM Wat-

son Knowledge Studio (WKS)[1] provide a rich set of document annotation tools along with the core component to train models based on annotated data. IBM WKS is based on the Statistical Information and Relation Extraction (SIRE) toolkit [Florian et al. 2004] for building trainable extractors for new applications and domains. SIRE provides components for mention detection using Maximum Entropy models [Berger et al. 1996] that can be trained from annotated data, a trainable co-reference component for grouping detected mentions in a document that correspond to the same entity, and a trainable relation extraction system.

The first step in using WKS is to define the type system. A type system represents the salient things in the target domain content that a human annotator or machine annotator should label with an annotation. The type system controls how content can be annotated by defining the types of entities that can be labeled and how relationships among different entities can be labeled. The annotator process manager typically works with subject matter experts in the domain to define the type system.

In this article, we use an O&G type system developed in an earlier project at our institution. It defines 31 entity types, drawn from the GeoSciML standard[2], and expanded with petroleum system and exploration concepts. The entity types can be broadly categorized as physical (earth materials, organic materials), geographical, geological including geological time, petroleum system, field development, and property/measurement. The type system also defines 653 relations between these entity types, such as 'formedDuring', 'causedBy', and 'composedOf'. This paper focuses on mention detection of the entity types and relations will not be further discussed.

This type system was used to annotate a set of scientific documents in the O&G domain; these serve as a 'golden set' against which different information extraction methods can be compared.

3. The 'golden set' from human annotations

The source documents for the 'golden set' (GS) were randomly selected from a corpus of 1298 publicly available English language geological reports, published by the United States Geological Survey (USGS), Geological Survey of Canada (GSC), and British Geological Survey (BGS). 155 text passages relevant to petroleum systems were extracted from the selected documents and annotated with entities and relations from the type system. Multiple occurrences of the same entity in a document were annotated as co-references.

The documents were annotated by a team of individuals with a background in geology, all with oil industry experience. In total, $38,322$ mentions of the 31 entity types were annotated. Inter-annotator agreement for entity mentions reached 0.84, and documents annotated by more than one annotator were adjudicated to arrive at a final version. Despite document cleaning, some documents contained text recognition noise.

The goal of the annotation was to gather training data to build WKS models to extract information about the entities and the relationships between them. The guidelines followed by the annotators were designed to minimize noise in the extracted data, and to

[1] `http://ibm.co/2kDFWph`

[2] `http://schemas.geosciml.org/`

focus on real-world information. Considering the examples below:

(1) The Wargal has produced oil at the Dhurnal f.eld.

(2) The Point Thomson and Kavik accumulations seem anomalous given their hydrocarbon phases and maturity levels.

(3) The USGS estimated a mean of 19.10 million bar-rels of oil, 50.585 trillion cubic feet of gas (TCFG), and 148.37 million barrels of natural gas liquids of undiscovered resources.

(4) Depth to production in the Aspen, Frontier, and Bear River in these *field* ranges from 100 to 2,000 ft, and oil gravity ranges from 22°to 48°API.

The annotations have the following characteristics:

- Entities containing text extraction noise such as "f.eld" for "field" in Example 1, or "bar-rels" for "barrels" were not annotated.
- Only mentions of real-world physical entities and their properties were annotated. Abstract concepts and definitions (e.g. "a reservoir is ...") were not annotated. One has to be careful here. We didn't just do named entities - we did nominals. For example, one might annotate "fault" talking about a specific fault, not just the San Andreas Fault.
- Annotators were able to use their background and context knowledge in deciding the type of an entity. For example, if the preceding document context made it clear that "Point Thompson" and "Kavik" in Example 2 are oil or gas fields, as opposed to rock formations, basins or geographical areas, then they would be annotated as `FIELD` even though the sentence has no explicit use of any variation of the word 'field'.
- Units and abbreviations are included in the annotations of measurements where present. A specific set of 43 properties are included, and all measures of other properties are left unmarked. Annotators were instructed to mark ranges of values expressed as "from X to Y ft" with separate mentions for the low and high ends of the range, to facilitate downstream processing of the information. Ranges expressed with a dash, in the form "x-y ft" were annotated as a single mention.

Annotations are provided for ungrammatical sentences. Notice the strange syntactic structure of sentence 4, including the wrong singular of "field". These GS annotations reflect the complexities of the real world and its linguistic encoding. Recovering these annotations poses a significant challenge to automated systems. In Section 5, we will take these 155 as a golden set and apply our pipeline to measure the difficulty of recognizing the human annotated entities. We focus our analysis on the `FIELD` entity type, which represents oil and gas fields. The GS contains 918 annotated mentions of `FIELD`.

4. A rule-based NLP pipeline

Our rule-based pipeline is composed of two main macro processes: linguistic analysis and fact extraction via Prolog rules. For the linguistic analysis we are using a combination of: 1) sentence segmentation; 2) tokenization, POS tagging, named entities recognition and parsing using English Slot Grammar (ESG, described in Section 4.1); and 3) a graph based word sense disambiguation (WSD).

We use the Apache OpenNLP [3] for sentence segmentation. OpenNLP implemented a supervised method for text segmentation using a model trained with the Bosque Corpus [Freitas et al. 2008]. The UKB [Agirre and Soroa 2009] algorithm performs an alignment between the words in the text and any semantic lexical database, in particular, we are using the Princeton English Wordnet (PWN, [Fellbaum 1998]). UKB is an implementation of the graph-based method for WSD. Graph-based techniques find and exploit the structural properties of the graph underlying the PWN (or any other lexical resource). Because the graph is analyzed as a whole, these techniques have the property of being able to find globally optimal solutions, given the relations between entities. Graph-based WSD methods are particularly suited for disambiguating word sequences, and they manage to exploit the interrelations among the senses in the given context.

Similar to [Fodor et al. 2008], we combine the linguistic analysis into a set of Prolog facts over sentence ids and token ids. For example the POS tag of a token is converted into `nlp_POS(s,i,POS)`, where `s` is the sentence id, `i`, the token id, and `POS` is the POS tag, represented as a string. Dependency relations between tokens are converted into three-argument predicates that range over tokens of a single sentence, for example: `nlp_conj(s,i1,i2)`. Here, `s` is the sentence id, and `i1` and `i1` are token ids. This works since there is no cross-sentence relation, but might need to be revisited once this support is added. Also, on a more practical note, using different predicates for the different dependency types (as opposed to a single relation where the dependency type is given as another parameter) allows a Prolog interpreter to properly index those predicates and thus having a large corpus of facts will not slow down the processing of rules later on. Next we apply several detection rules (Section 4.2) that match patterns on the facts to enhance the parse tree.

4.1. English Slot Grammar

English Slot Grammar (ESG) [McCord 1990] is a deep parser in the sense that the parse trees it produces for a sentence (or segment) show a level of logical analysis (deep structure). However, each parse tree also shows a surface-level grammatical structure (surface structure), along with the deep structure. The parse trees for a segment are ranked according to a parse scoring system. The system is divided into a large language-universal shell and language-specific grammars for English, German, French, Spanish, Italian, and Portuguese. The main steps of SG parsing are (A) tokenization and segmentation, (B) morpholexical analysis, and (C) syntactic analysis. Unlike some parsers, SG uses no part-of-speech (POS) tagger; the corresponding information simply comes out of syntactic analysis.

As the name suggests, Slot Grammar is based on the idea of slots. Slots have two levels of meaning. On the one hand, slots can be viewed as names for syntactic roles of phrases in a sentence. On the other hand, certain slots (complement slots) have a semantic significance. They can be viewed as names for argument positions for predicates that represent word senses. Figure 1 shows an example of slots, and the phrases that fill them. It shows for example that 'Wargal' fills the subj (subject) slot for the verb 'has', 'Blue Creek' fills the nadj slot for the word 'field' etc. One can see then that the slots represent syntactic roles.

[3] `https://opennlp.apache.org`

```
  .----- ndet          the1(1)              det sg def the ingdet
.------- subj(n)       Wargal(2)            noun propn sg notfnd
o------- top           have_perf(3,2,4)     verb vfin vpres sg vsubj  auxv
`------- auxcomp(ena)  produce1(4,2,5,6)    verb ven vcreate (nform production)
  `----- obj(n)        oil1(5,u)            noun cn sg physobj massn ent material
  `----- comp(p)       at1(6,4,9)           prep pprefv staticp
    | .- ndet          the1(7)              det sg def the ingdet
    | .- nadj          Blue Creek(8)        noun propn sg capped notfnd
    `--- objprep(n)    field1(9,u,u)        noun cn sg location geoarea ent
```

Figure 1. ESG parse of "The Wargal has produced oil at the Blue Creek field."

To illustrate the semantic view of slots, consider that there is a word sense of 'produce' which, in logical representation, is a predicate, say $produce_1$, where $produce_1(e,x,y,z)$ means "e is an event were x produces y at z". Slots that represent predicate arguments in this way are called complement slots. Such slots are associated with word senses in the Slot Grammar lexicon - in slot frames for the word senses. All other slots are called adjunct slots (e.g. 'ndet').

Given this dual role of slots, Slot Grammar parse trees show two levels of analysis – the surface syntactic structure and the deep logical structure. The two structures are shown in the same parse data structure. So on each line of the parse display, you see a head word sense in the middle section, along with its logical arguments. To the left of the word sense predication, you see the slot that the head word (or node) fills in its mother node, and then you can follow the tree line to the mother node. To the right, you see the features of the head word (and of the phrase which it heads). The first feature is always the part of speech (POS). Further features can be morphological, syntactic, or semantic. For instance, Figure 1 shows that 'oil' was recognized as a material and a mass noun while 'field' was recognized as a geoarea and location. The semantic features are more open-ended, and depend on the ontology and what is coded in the lexicon.

Regarding the arguments given to word sense predicates in the parse, the first argument is just the node index, which is normally the word number of the word in the sentence. This index argument can be considered to correspond to the event argument (with a broad interpretation of 'event'). The remaining arguments correspond to the complement slots of the word sense – or rather to the fillers of those slots. They always come in the same order as the slots in the lexical slot frame for the word sense. So for a verb, the first of these complement arguments (the verb sense's second argument) is always the logical subject of the verb. Generally, all the arguments are logical arguments.

By using ESG, we are able to generate amalgamated tokens such as 'Blue Creek' that are treated as a single word in the next steps of the pipeline, thus possibly simplifying not only the dependency analysis, but also the necessary rules. We feel that this is a healthy approach from an engineering point of view, as this makes the rules much simpler to implement and maintain, as well as making the best use of what each NLP tool has to offer. This process is not only used for named entities but also for other multi-word expressions such as 'more than'. In one of its many options for names detection, ESG recognize sequences of capitalized words as multi-word proper nouns, taking into account some functional words, the dictionary entries and capitalization at sentence beginnings. Nevertheless, ESG fails sometimes, as we will discuss in the following sections.

4.2. Prolog rules

A subset of the implemented Prolog rules for handling mentions of type `FIELD` follows. Recall from Section 4.1 that we expect all proper names to have been retokenized into single tokens, thus simplifying the complexity of the rules. The rules from Figure 2 rely on the existence of an *anchor word* that will give semantic meaning to the words connected to it. The basic idea (lines 14-20 in Figure 2) is to find proper nouns that are connected to this anchor word via a `nadj` dependency (noun adjunct slot) which will indicate that this word is a field name (e.g. 'Blue Creek field' in Figure 1).

```
1  connected_to_anchor(S,ConnectedNouns,AnchorLemma) :-
2          nlp_lemma(S,AT,AnchorLemma),
3          nlp_nadj(S,EntryPoint,AT),
4          graph_conj(S,G),
5          reachable(EntryPoint,G,ConnectedToEntry),
6          Tmp1 = [EntryPoint|ConnectedToEntry],
7          exclude(cord(S),Tmp1,Tmp2),
8          include(propn(S),Tmp2,ConnectedNouns).
9
10 anchor(S,[X],A) :-
11         connected_to_anchor(S,CN,A),
12         member(X,CN).
13
14 anchor(S,[X],A) :-
15         propn(S,X),
16         nlp_lemma(S,AT,A),
17         nlp_nadj(S,X,AT).
18
19 field(S,TL) :-
20         anchor(S,TL,'field').
21
22 basin(S,TL) :-
23         anchor(S,TL,'basin').
```

Figure 2. Example of rules

A more complex version of this rule would use synonyms and hypernyms of the world 'field' in line 4, like 'oilfield', 'gas field'. A more concise solution would be to use *senses* related to oil fields and thus capturing all possible words related to them. The word 'field' has 17 senses in PWN, but, unfortunately, in the word sense disambiguation provided by UKB, the most frequent sense selected was {`05996646-n`: a branch of knowledge} (510 times) followed by the sense {`14514039-n`: a particular environment or walk of life} (64 times). The two expected senses were never selected. The first one is the {`08659446-n`: a geographic region (land or sea) under which something valuable is found.}, and the second one, its hyponym, {`08659861-n`: a region rich in petroleum deposits (especially one with producing oil wells)} which doesn't contain the word 'field' but only 'oilfield', justifying the result. This is one of the gaps in PWN (see other cases in Section 7) that we are solving with its adaptation to the O&G domain [Muniz et al. 2018].

We can also handle more complex phrasal structures, such as conjunctions (lines 1-8). The predicates `graph_conj` and `reachable` are used to handle sentences such as "Active exploration is now focused in Blue Creek, White Oak Creek, and Short Creek fields in the northern part of the basin". ESG annotates conjunctions using a linked list style (see Figure 3) and the rules have to collected two or more elements in coordination.

We have also rules (not presented in Figure 2) to deal with more complex struc-

```
...
`---------------- vprep      in1(6,3,108)          prep staticp
  | .----------- lconj      Blue Creek2(8)        noun propn sg location
  `-+----------- objprep(n) ,(108)                noun cn pl location cord
    |   .------- lconj      White Oak Creek3(11)  noun propn sg location
    | .-+------- nadj       and1(12)              noun propn pl location cord
    | | `------- rconj      Short Creek4(14)      noun propn sg location
    `-+--------- rconj      field1(15,u,16)       noun cn pl location geoarea
...
```

Figure 3. The fragment of an coordination.

ture. For example, "Fields that are along this zone of low percent sulfur are the Bretana, Dorrisa, Huayuri, Huayuri Sur, Sun, Tetete, and Valencia". Here, we have a copula followed by several conjunctions where the word 'fields' is the subject of the copula.

In practice we have eight rules: six rules to deal with three different types of conjunction, and two rules for single compounds. These rules are reused (lines 22-23) for any type of entity that is associated with an anchor word (basins, wells etc.).

5. The experiment

In this section we describe the main experiment we conducted. We took the 155 scientific articles manually annotated by specialists in the O&G domain (the golden set) with the type system described in Section 2 and ran them in our rule-based pipeline described in Section 4. The idea is to evaluate the performance of our rule-based method in detecting all entities mentioned and annotated by humans in the documents. We have focused our analysis on comparing the ability of our pipeline to detect the mentions of type `FIELD`. At this time, we have not addressed the identification of relations.

In the golden set documents, there are 918 annotations of type `FIELD`. Of those, 489 are variants of the word "field". Given that, the remaining 429 annotations are potential names of fields. After removing known suffixes such as "field", "oil field", and "gas field" we have 239 distinct names for fields in the golden set.

Running the ruleset over the 155 documents produces 109 distinct field names, 102 of which match the golden set. This gives a precision of .94 and recall of .43 (F1=.595). The statistical model trained with the annotated documents achieved precision of .62 and recall of .76 for the `FIELD` type. However, this value includes all 918 `FIELD` annotations, so cannot be directly compared.

6. Some qualitative analysis

Our approach for information extraction relies on a robust linguistic analysis. In this section we present some evaluation in sentence segmentation, parsing, and word sense disambiguation that we encountered in the technical documents. Dealing with scientific articles impose many difficults. Since most of the text passages were retrieved from PDF files, non ASCII characters generate garbage characters (i.e. "$10^6/\Delta ft/s$"). We have to manually clean the files removing around 20% of the sentences that contain some unknown symbol or broken words.

Regardings the recognition of sentences bounderies, for better evaluate the parsing, we manually fixed the sentence splitting step. We found around 100 cases of erro-

neous splitting of sentences caused by uncommon abbreviations such as “unpub. data”, “[...] is located in sec. 29, T. 25 N., R. 91 W.” or “fig. 38” and citations.

One of the most problematic issues for the parser is the detection of multiword expressions (MWE). MWEs such as “depositional environments”, “clastic units”, “depositional basin” and “shoreface sandstone” were generally recognized as constructions where the first word is an adjective modifying a noun but they are actually lexicalized or institutionalized phrases in the O&G domain [Sag et al. 2002]. Conversely, we also have constructions that are not MWE but are recognized as so: “gas liquids”, “reservoir objectives” etc. Sentence 5 shows an error in the analysis of the expression “vitrinite reflectance”, which should be considered an MWE (a method for measuring the maturity of the rock), but it was tagged as separated nouns “vitrinite” and “reflectance”, generating the wrong analysis and a nonsense interpretation that “reflectance data” is “vitrinite”.

(5) At the northern end of the section, however, vitrinite reflectance data from the Eocene and Oligocene sections are from higher stratigraphic positions and indicate only small amounts of previous burial.

MWEs that are proper nouns also pose a challenge to the parser in several ways, since the amalgamation of the tokens is not uniform. In the sentences 6 and 7, ESG correctly identified the proper nouns ‘Molina’ and ‘Piceance’ despide the capitalization that would suggest the terms ‘Member’ and ‘Basin’ as part of the names. On the other hand, ESG erroneously broke the proper nouns ‘Piceance Creek Dome’ and ‘Sulphur Creek’ given that their parts are also in the ESG dictionary as common nouns. In the first case, only the word ‘Piceance’ is consider a proper noun.

(6) At the Piceance Creek Dome field in the central part of the Piceance Basin . . .

(7) The Molina Member and “Wasatch G” sandstone reservoirs produce gas at Piceance Creek Dome and Sulphur Creek fields in the central part of the Piceance Basin . . .

Since every mistagged word has a cascading effect on the syntactic analysis of the phrase, these errors suggested us to implement workarounds in our rules to consider both types (common and proper nouns), which is not ideal. On the other hand, the addition of entity names in the ESG lexicon is easy and directly impacts the results. A method for proper names acquisition from corpora is necessary and some previous work on combination of named entities recognition with deep parsing was already developed [Teufel and Kan 2011, Copestake et al. 2006].

There were also instances where no anchor word was present, as in example 8. Here, the human annotator likely used background knowledge or the preceding document context to infer that those highlighted names were mentions of fields.

(8) In the deepest parts of the province at the Adobe Town, Eagles Nest, and Wagon Wheel locations (figure 5B), the generation of gas from the cracking of oil began at about 56 Ma, within about 6 m.y.

(9) Outside this arcuate trend, cumulative production exceeding 300 MMCFG of gas has a patchy distribution; in the Little Sandy Creek, Moundsville, Little Buck Creek, and Taylor Creek fields, for example, only one well has produced more than 100 MMCFG (Pashin and others, 2004).

(10) The eight oil fields in the upper Sunniland Formation that have produced, or have EUR's, more than 1 MMBO are *Bear Island*, *Corkscrew*, *West Felda*, *Lehigh Park*, *Mid-Felda*, *Raccoon Point*, *Sunniland*, and *Sunoco-Felda*.

In one case, sentence 9 mentioned the 'Taylor Creek' field but was not annotated in the golden set. Our pipeline found it. More complex phrasal structures, as in sentence 10, were also supported (see Section 4.2). However some of the proper nouns were not identified properly.

The identification of numbers, ranges of numbers and quantities is also challenging. In the analysis of the sentence 11 we obtain two separated propositions with nothing directly relating the "trillion" word to the numbers.

(11) Nonassociated gas resources range between 23.9 and 44.9 trillion cubic feet (TCF) (95% and 5% probabilities), with a mean of 33.3 TCF.

(12) Mavor and others (2003) report TOC varies from 0.5 to 2.5 percent with an average TOC of 1.3 percent.

Citations are very common in scientific articles (approx. 10 per document in the GS). The sentence 12 has it citation analysed as a conjunction by ESG, Figure 4, making the post-processing much more complicated than necessary. Unfortunately, adapting ESG to handle citations as MWEs, amalgamating the words, is not trivial. In the future, we are considering a pre-processing step to deal with citations before the parser.

```
  .--------- lconj       Mavor(1)               noun propn sg capped notfnd
.-+--------- subj(n)     and1(2)                noun cn pl detr cord
| | .------- lconj       other1(3,u)            noun cn pl detr
| `-+------- rconj       ((103)                 noun cn pl detr cord yr
|   `------- rconj       2003(4,u)              noun num sg pl sgpl yr
o----------- top         report2(5,2,7,u,u,u,u) verb vfin vpres pl vsubj sayv
...
```

Figure 4. Fragment of a citation.

7. Pipeline evaluation

For better evalutation of our pipeline, we have experimented with other tools comparing the results of intermediate steps. For parsing, we compared ESG to the statistical-based parser UDPipe [Straka and Straková 2017] and the open-source HPSG [Pollard and Sag 1994] grammar for English [Flickinger 2000]. For word sense disambiguation, we compared UKB to JIGSAW [Basile et al. 2007].

UDPipe is a trainable pipeline for tokenization, tagging, lemmatization and dependency parsing of CoNLL-U files. UDPipe is language-agnostic and can be trained given annotated data in CoNLL-U format. UDPipe is on the top 10 best parsers in the last

UD shared task. [4] We are using the UDPipe dependencies model trained with the English corpus released by the Universal Dependencies project, version 2.2 [Nivre 2018]. Universal Dependencies (UD) [Nivre et al. 2016] is a recent approach to dependency parsing that tries to maximize the sharing of structures between languages. UD has a set of dependency labels and POS tags that are designed to fit many languages, and a series of annotation manuals that guide treebank builders to use the labels and tags in a uniform way. The advantages of using UD compared with the ESG annotation are: (1) it is being widely adopted as a *standard* dependencies schema with comprehensive documentation and examples; and (2) some of its principles, such as, "the primacy of content words" [Nivre et al. 2016] facilitate the task of information extraction.

Similar to UKB, the JIGSAW algorithm [Basile et al. 2007] disambiguates each (noun, verb, adjective, adverb) word of the sentence by assigning it the sense with the highest probability. Different from UKB, JIGSAW exploits the WordNet senses and uses a different disambiguation strategy for each part of speech, taking into account the context of each word.

The parsing task is notably the most important part of our pipeline. To evaluate the accuracy of ESG, we first compared it to ERG. In general, grammar-based parsers produce many parse trees for a sentence. During the experiment, we asked both tools to give us the best of the possible trees. Recall that the corpus contains 5,591 sentences with an average of 28 words per sentence. Most of the sentences contain between 10 and 40 words. ESG was more robust, parsing 4,770 sentences (85%) compared to ERG that parsed only 3,528 sentences (63%). In total, both parsers failed to parse 517 (9%) sentences and 3,224 sentences (58%) were parsed by both tools. In contract, statistical parsers usually produce one parse tree for every input, even when the analysis does not make sense at all. Given that, for evaluating the relative performance of the grammar-based parsers compared to the statistical parser, we would need to manually compare each parse tree. But for a first approximation, we opted to use an evaluation tool that, although still in the first stages of development, tries to detect possible inconsistencies in the syntactic analysis given the UD guidelines formalized in an ontology [Paulino Passos 2018]. This tool detected 2,593 sentences with at least one possible error (e.g. a token verb cannot be the head of another token in an 'appos' dependency relation), which gives us an approximation for the performance of UDPipe: 3,005 sentences (53%) seems to be parsed in a meaningful way by UDPipe. So far, we are talking about the capability of the parsers to produce a parse tree for a given sentence, in respect of time spent for the analysis, ESG processed the corpus in 1.26 minutes, UDPipe took 1.51 minutes, and ERG needed approximately 3 hours.

From the results presented in the previous paragraph, we can conclude that ESG has not only a high-grade performance but it is the best option between the alternatives presented here. Nevertheless, ERG is a product of a consortium. The partners have adopted HPSG and Minimal Recursion Semantics (MRS), two advanced models of formal linguistic analysis. They have also committed themselves to a shared format for grammatical representation and to a rigid scheme of evaluation, as well as to the general use of open-source licensing and transparency. That means, syntactic and semantic representations are well documented and modular, theoretical results have many implementations

[4] `http://universaldependencies.org/conll17/results.html`

that learn from each other. For instance, one can use more than one parser with the ERG grammar. Each parser has its own capabilities. in particular, the combination of statistical POS tagger with deep parsing makes the parser PET [Callmeier 2000] a very attractive tool for solving many issues related to recognition of MWEs described here. In contrast, ESG techniques and theoretical principles are implemented in a single tool, developed over the years by a tiny group of experts. Not all decisions are well documented and the interaction with other tools is not easy. For instance, although the lexicon is easy to expand, it is not easy to deal with citations without affecting the parser algorithm. Finally, the low quality of the statistical parser is easy to explain, the model used was trained with corpus from a News domain, with a very different lexicon and style. The problem is the cost to annotate a domain specific corpus for training a better model.

Finally, to evaluate the WSD we opted to check the aggrement between UKB and JIGSAW. As expected, after the elimination of the stop words (e.g. prepositions, determinants) not present in PWN, we found many domain specific words missing in PWN. Considering the top most frequent ones by part-of-speech:

adjectives 'stratigraphic' (171), 'eocene' (86), 'permian' (76) and 'paleocene' (67) .
nouns the abbreviations 'bcfg' (115), 'tps' (105), and 'mmbo' (105) and the words 'facies' (112), 'mudstone' (91), 'anticline' (83), and 'ellesmerian' (81) .
verbs 'rift' (21), 'overmature' (7), 'overpressure' (6), and 'recomplete' (5).
adverbs 'termally' (33), 'unconformably' (23), 'stratigraphicly' (22), and 'seismically' (6).

Table 1 summaries some of our finds by lemmas aggregated by part-of-speech. In the second and third columns we show the number of times UKB/JIG aggre (eq, percent relative to column freq) and disagree (neq). In the column 'freq' we show the number of tokens. Column 'sum senses' is the sum of the number of senses in PWN for each lemma. Column 'mean senses' is the average of senses per lemma. Finnaly, the column 'distinct' shows the number of distinct lemmas. That is, the first line says that we found 10,225 tokens annotated as nouns (1,898 distinct lemmas) that have at least one sense in PWN. The average of senses per lemma for nouns is 3.54 and the total number of senses for all lemmas of nouns from the corpus in PWN is 33,629.

pos	eq	neq	freq	sum senses	avg senses	distinct
n	23404 (69%)	10225	33629	6724	3.54	1898
v	10619 (77%)	3154	13773	4121	5.79	711
adv	2816 (87%)	391	3207	612	1.84	332
a	9422 (82%)	2067	11489	2654	3.17	837

Table 1. Numbers of WSD results by lemma.

It is well-known that the verbs have high polysemy. In PWN, for example, the average number of senses for verbs is 2.17 (with 36 verbs with over 20 senses), and for nouns is 1.22 (with only five nouns with over 20 senses). Despide that, Table 1 helps us to conclude that in this corpus, given the restricted number of different verbs used by the authors of scientific articles, the WSD of nouns is amost 10% harder than the WSD of verbs. This could, for instance, justify a different approach for WSD, based on the idea of *selectional restrictions* [Allen 1995] combined with UKB.

8. Conclusion and future work

For information extraction tasks, approaches that rely exclusively on human annotations can be prohibitively expensive and the required experts may not always be available. Human annotators require a significant investment in guidance, coordination, and evaluation. This paper has described a workbench for linguistic and rule-based information extraction. We applied natural language processing tools trained on the news domain to 155 annotated text passages from geological reports. Our pipeline includes sentence segmentation, parsing, word sense disambiguation. We developed a set of rules that matched the linguistic annotations to recognize entities. This approach can provide standalone information extractor components that do not require expensive training data, and can help *pre-annotate* texts prior to human annotation[5].

In our experiment with oil field names, our pipeline emits strings that were found to be field names, and we measured the system performance against distinct string values after a normalising step that removes well known suffix variations such as (*field* and *gas field*), achieving a precision of .94 and recall of .43 (F1=.595) without supervised learning. Future work will include *mentions* found against those that were annotated.

Rule-based approaches are well known and explored in several other projects [Chiticariu et al. 2010, Fagin et al. 2015], but we feel that there is opportunity for improvement by mixing several different pipelines (to use what each pipeline can contribute better) and taking advantage of recent developments in dependency annotations [Schuster and Manning 2016, Stanovsky et al. 2016, Reddy et al. 2016]. A better software engineering approach to rules can lead to significant reduction in complexity in managing the rule base.

Our experiments also reveal challenges in integrating linguistic information from different NLP pipelines. For statistical based modules, such as POS tagging and dependency parsing, the input tokens should be compatible with the training data. Multiword expressions could be improved through better word sense disambiguation. Similarly, numbers, dates and quantities are recognized in different ways across NLP pipelines.

We conclude that despite these challenges, high quality NLP tools developed and tested on data from other domains can be adapted for entity extraction in technical domains without requiring domain-specific supervision.

References

Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Allen, J. (1995). *Natural language understanding*. Pearson.

Antoniak, M., Dalgliesh, J., Verkruyse, M., Lo, J., et al. (2016). Natural language processing techniques on oil and gas drilling data. In *SPE Intelligent Energy International Conference and Exhibition*. Society of Petroleum Engineers.

Basile, P., De Gemmis, M., Gentile, A. L., Lops, P., and Semeraro, G. (2007). Uniba: Jigsaw algorithm for word sense disambiguation. In *Proceedings of the 4th International*

[5]This is partially supported today with dictionaries and basic regular expression types of rules.

Workshop on Semantic Evaluations, pages 398–401. Association for Computational Linguistics.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Callmeier, U. (2000). Pet–a platform for experimentation with efficient hpsg processing techniques. *Natural Language Engineering*, 6(1):99–107.

Chiticariu, L., Krishnamurthy, R., Li, Y., Raghavan, S., Reiss, F. R., and Vaithyanathan, S. (2010). Systemt: An algebraic approach to declarative information extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 128–137, Stroudsburg, PA, USA. Association for Computational Linguistics.

Copestake, A., Corbett, P., Murray-Rust, P., Rupp, C., Siddharthan, A., Teufel, S., and Waldron, B. (2006). An architecture for language processing for scientific texts. In *Proceedings of the UK e-Science All Hands Meeting 2006*.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*. Morgan and Claypool.

Fagin, R., Kimelfeld, B., Reiss, F., and Vansummeren, S. (2015). Document spanners: A formal approach to information extraction. *J. ACM*, 62(2):12:1–12:51.

Feblowitz, J. (2013). Analytics in oil and gas: The big deal about big data. In *SPE Digital Energy Conference and Exhibition*, The Woodlands, Texas.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Flickinger, D. (2000). On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Florian, R., H, H., Ittycheriah, A., Jing, H., Kambhatla, N., Luo, X., Nicolov, N., and Roukos, S. (2004). A statistical model for multilingual entity detection and tracking. IBM T.J. Watson Research Center.

Fodor, P., Lally, A., and Ferrucci, D. A. (2008). The prolog interface to the unstructured information management architecture. *CoRR*, abs/0809.0680.

Freitas, C., Rocha, P., and Bick, E. (2008). Floresta sintá (c) tica: bigger, thicker and easier. In *International Conference on Computational Processing of the Portuguese Language*, pages 216–219. Springer.

McCord, M. C. (1990). Slot grammar. In *Natural language and logic*, pages 118–145. Springer.

Muniz, H., Chalub, F., Rademaker, A., and de Paiva, V. (2018). Extending wordnet to geological times. In *Global Wordnet Conference 2018*, Singapore. to appear.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In Chair), N. C. C.,

Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Nivre, J. e. a. (2018). Universal dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Palkowsky, B. (2005). A new approach to information discovery. In *SPE Annual Technical Conference and Exhibition*, Dallas, Texas. Society of Petroleum Engineers.

Paulino Passos, G. (2018). A formal specification for syntactic annotation and its usage in corpus development and maintenance: A case study in Universal Dependencies. Master's thesis, UFRJ/COPPE/PESC. (Submitted).

Pollard, C. and Sag, I. A. (1994). *Head-driven phrase structure grammar*. University of Chicago Press.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

Reddy, S., Täckström, O., Collins, M., Kwiatkowski, T., Das, D., Steedman, M., and Lapata, M. (2016). Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword Expressions: a pain in the neck for NLP. In *Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15, Heidelberg. Springer Berlin.

Schuster, S. and Manning, C. D. (2016). Enhanced english universal dependencies: an improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Stanovsky, G., Ficler, J., Dagan, I., and Goldberg, Y. (2016). Getting more out of syntax with props. *CoRR*, abs/1603.01648.

Straka, M. and Straková, J. (2017). 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*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Teufel, S. and Kan, M.-Y. (2011). Robust argumentative zoning for sensemaking in scholarly documents. In *Advanced Language Technologies for Digital Libraries*, pages 154–170. Springer.

Tohidi, H., Ibrahim, H., and Murad, M. A. A. (2014). Improving named entity recognition accuracy for gene and protein in biomedical text literature. *International journal of data mining and bioinformatics*, 10(3):239–268.