Template Extraction from unstructured Wikipedia text using NLP techniques.

1st Sameer Ahmed
Btech-IT)
IIIT Allahabad)
UP, India
iit2020053@iiita.ac.in

***Abstract*—This document describes a project on information extraction using template filling in natural language processing (NLP) and linguistics. The project involves extracting information from unstructured textual data such as Wikipedia articles and filling three given templates for buying events, job title relations, and part-of relations. The proposed solution for the project focuses on a rule-based approach, which involves defining a set of syntactical and grammatical rules for natural language and using various techniques such as named-entity recognition, co-reference resolution, semantic-role labeling, dependency parsing, phrase matching, and wordnet features to extract information and fill the templates. The document provides an example sentence for each template and outlines the steps involved in extracting the relevant information to fill the templates.**

***Index Terms*—NLP, EventExtraction,Wikipedia,Lemmatization,Preprocessing,Templates**

I. Introduction

A. Problem Description: The task is to perform information extraction using template (slot) filling on unstructured textual data, such as Wikipedia articles, by applying various NLP techniques. The goal is to extract information and fill three given templates related to buying events, job title relations, and part-of relations between locations. 30 Wikipedia articles are provided, 10 each related to organizations, persons, and locations.

B. Proposed Solution: The proposed solution involves a rule-based approach, where a set of syntactical and grammatical rules are defined for a natural language to fill the templates. The following techniques are used: Named-Entity Recognition (NER) and Co-reference Resolution to resolve pronoun references. Sentencizer to find sentence boundaries and extract standalone sentenceLemmatization, NER, and Semantic-Role Labelling (SRL) to extract information for Template1 (BUY). Lemmatization, NER, SRL, Dependency Parsing, and Phrase Matching to extract information for Template2 (WORK). Lemmatization and Wordnet Features to extract information for Template3 (PART). An example is given for each template to illustrate the approach, where the proposed techniques are applied in sequence to extract the required information from the example sentence. The extracted template is presented at the end of each example.

Dependency Parsing: We perform dependency parsing on the sentence to identify the relationship between words and to extract the verb associated with the noun “works” and the related noun “position”. Phrase Matching: We use regular expressions to match patterns that represent job titles or positions. Extracted Template: WORK(“Steven Paul Jobs”, “Apple Inc.”, “chairman; chief executive officer (CEO); co-founder”, “”) For Template3(PART): Example Sentence: Dallas is a technical hub in Texas.Lemmatization: We find lemmas of all the tokens in the sentence. extract buying events, job titles, and part-of relations. Wordnet features: We use Wordnet features to identify the hypernym of the entity and find its root hypernym to extract the part-of relation. Extracted Template: PART(“Dallas”,”Texas”)

C.Conclusion: In this project, we proposed a rule-based approach for information extraction using template filling. We used multiple techniques such as Named-Entity Recognition, Co-reference Resolution, Lemmatization, Semantic-Role Labelling, Dependency Parsing, Phrase Matching, and Wordnet Features to extract information from the given unstructured text and fill the provided templates. We applied this approach on the given Wikipedia articles related to Organizations, Persons, and Locations to Problem: NER discrepancies: spacy’s v2.2.0 detected ”Souq.com” as ”ORG” but not v2.2.5, while ”Ring” was detected as ”ORG” in v2.2.5 but not v2.2.0. Solution:Ensemble both models to maximize entity detection coverage.Job title labelling: used open-source data and fed it into spacy’s EntityRuler, which matches provided patterns and labels entities as ”TITLE”.Tool selection: used AllenNlp for Semantic Role Labelling and Coreference Resolution, and Spacy for other tasks. WORK Template extraction: used both ”ORG” and ”PERSON” labels, then resolved false positives using dependency parsing. Potential Improvements: Train own models instead of relying on default ones. Continuously search for areas where the proposed solution may fail and provide workarounds.

II. Literature Survey

A. Paper 1

Author– Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov ·
Title– Bag of Tricks for Efficient Text Classification
Approach– text classification. Our experiments show that our fast text classifier fastText is often on par with deep learning classifiers in terms of accuracy, and many orders of magnitude faster for training and evaluation. We can train

fastText on more than one billion words in less than ten minutes using a standard multicore CPU, and classify half a million sentences among 312K classes in less than a minute

Performance – We run fastText for 5 epochs and compare it to Tagspace for two sizes of the hidden layer, i.e., 50 Model prec@1 Running time Train Test Freq. baseline 2.2 - - Tagspace, h = 50 30.1 3h8 6h Tagspace, h = 200 35.6 5h32 15h fastText, h = 50 31.2 6m40 48s fastText, h = 50, bigram 36.7 7m47 50s fastText, h = 200 41.1 10m34 1m29 fastText, h = 200, bigram 46.1 13m38 1m37 Table 5: Prec@1 on the test set for tag prediction on YFCC100M. We also report the training time and test time. Test time is reported for a single thread, while training uses 20 threads for both models. and 200. Both models achieve a similar performance with a small hidden layer, but adding bigrams gives us a significant boost in accuracy. At test time, Tagspace needs to compute the scores for all the classes which makes it relatively slow, while our fast inference gives a significant speed-up when the number of classes is large (more than 300K here). Overall, we are more than an order of magnitude faster to obtain model with a better quality. The speedup of the test phase is even more significant (a 600× speedup). Table 4 shows some qualitative examples. 4 Discussion and conclusion In this work, we propose a simple baseline method for text classification. Unlike unsupervisedly trained word vectors from word2vec, our word features can be averaged together to form good sentence representations. In several tasks, fastText obtains performance on par with recently proposed methods inspired by deep learning, while being much faster. Although deep neural networks have in theory much higher representational power than shallow models, it is not clear if simple text classification problems such as sentiment analysis are the right ones to evaluate them. We will publish our code so that the research community can easily build on top of our work

Dataset– YFCC100M dataset
Year– 2016

B. Paper 2

Author– Prit Thakkar Ronit Patel·
Title– Template based Information Extraction System using NLP Techniques
Approach– The task of Information Extraction (IE) based on template filling can be performed using multiple approaches. They are listed below: • Supervised • Semi-supervised • Rule-based Approach For this project we will be basically focusing on the Rule-based Approach, where we will be defining a set of syntactical and grammatical rules of a natural language and then use them to fill our templates for the problem. After reviewing multiple research paper on Rule- based Information Template extraction, we found multiple techniques to achieve this task. Some of them are listed below. • Using Wordnet features of tokens from the text • Semantic-Role Labelling(SRL) • Dependency Parsing • Constituency Parsing • Named-Entity Recognition(NER) • Co-reference Resolution • Phrase Matching
Performance –We faced few problems related to NER for example, spacy's v2.2.0 model detect entity "Souq.com" as "ORG" but spacy's v2.2.5 model doesn't. Same ways entity "Ring" was detected as "ORG" in v2.2.5 but not in v2.2.0. Thus, we decide to ensemble both models to get maximum coverage of detecting entities. • Also, spacy's NER doesn't provide labelling to Person's Job Title entities. So, we used open-source data for Job Title and then provided it as an input to spacy's EntityRuler which after applying default NER matches patterns provided to it and labels matched entities as "TITLE". • There were many 3rd party NLP tools that are available for python. And the problem was some provide better results for certain tasks than the other. So, after detailed analysis we concluded to use AllenNlp for Semantic Role Labelling and Coreference Resolution; and Spacy for rest of the tasks. • For the WORK Template extraction, spacy's NER tagged certain Organization as "PERSON" and so we used both "ORG" as well as "PERSON" label to trigger the WORK Template extraction. The given solution popped some False Positive which we resolve using dependency parsing.

Dataset– jobtitledataset.csv
Year– 2021

C. Paper 3

Author– Xinya Du, Claire Cardie
Title–Event Extraction by Answering (Almost) Natural Questions
Approach– The problem of event extraction requires detecting the event trigger and extracting its corresponding arguments. Existing work in event argument extraction typically relies heavily on entity recognition as a preprocessing/concurrent step, causing the well-known problem of error propagation. To avoid this issue, we introduce a new paradigm for event extraction by formulating it as a question answering (QA) task that extracts the event arguments in an end-to-end manner. Empirical results demonstrate that our framework outperforms prior methods substantially; in addition, it is capable of extracting event arguments for roles not seen at training time (zero-shot learning setting).
Performance –we introduce a new paradigm for event extraction based on question answering. We investigate how the question generation strategies affect the performance of our framework on both trigger detection and argument span extraction, and find that more natural questions lead to better performance. Our framework outperforms prior works on the ACE 2005 benchmark, and is capable of extracting event arguments of roles not seen at training time. For future work, it would be interesting to try incorporating broader context (e.g., paragraph/document-level context (Ji and Grish- man, 2008; Huang and Riloff, 2011; Du and Cardie, 2020) in our methods to improve the accuracy of the predictions.

Dataset– ACE 2005 corpus

Year– 2021

III. Methodology

The task of Information Extraction (IE) based on template filling can be performed using multiple approaches. They are listed below:

- Supervised
- Semi-supervised
- Rule-based Approach

For this project we will be basically focusing on the Rule-based Approach, where we will be defining a set of syntactical and grammatical rules of a natural language and then use them to fill our templates for the problem. After reviewing multiple research paper on Rule- based Information Template extraction, we found multiple techniques to achieve this task. Some of them are listed below.

- Using Wordnet features of tokens from the text
- Semantic-Role Labelling(SRL)
- Dependency Parsing
- Constituency Parsing
- Named-Entity Recognition(NER)
- Co-reference Resolution
- Phrase Matching

Initially, we apply Named-Entity Recognition and Co-reference Resolution to the given unstructured data in order to resolve pronoun references so that we can extract templates from a single standalone sentence. After that we apply Sentencizer to find sentence boundary to extract a single standalone sentence. Now for extracting information from the extracted sentence to fill Template1, we use Lemmatization, NER, and SRL. For extracting information to fill Template2, we use Lemmatization, NER, SRL, Dependency Parsing and Phrase Matching. And for Template3, we use Lemmatization and then extract Wordnet Features to find the part-of relation.

For Template1(BUY):

Example Sentence: In 2017, Amazon acquired Whole Foods Market for USD13.4 billion, which vastly increased Amazon's presence as a brickandmortar retailer.

1. Lemmatization: We find lemmas of all the tokens in the sentence.
2. NER: We will apply NER on the sentence to find out entities and categorize them according to their entity labels.
3. SRL: Now we perform SRL to extract semantic role information about the verb present in the sentence. We perform this step only if we find that the verb lemma into consideration with a sense similar to "buy", for this example it is "acquire".

Extracted Template: BUY("Amazon", "Whole Foods Market", "USD13.7 billion", "", "")

For Template2(WORK):

Example Sentence: Steven Paul Jobs was the chairman, chief executive officer (CEO), and co-founder of Apple Inc.

1. Lemmatization: We find lemmas of all the tokens in the sentence similarly as we did for Template1.
2. NER: We would similarly find NER as we did in Template1.
3. Dependency Parsing: We would use the dependency parsing of the sentence to fill this template.

As it is clearly visible that the entity with label "ORG" i.e Apple Inc.(labelled as a part of NER) is related to all the "Position" tokens which is related to the Auxiliary Verb whose child with dependency "nsubj" is an entity labelled as "PERSON".

Extracted Template: WORK("Steven Paul Jobs", "Apple Inc.", "chairman ; chief executive officer (CEO); co-founder", "")

But there are cases which are not covered with the dependency parsing technique. So, we use the ensemble approach using SRL and Phrase Matching.

Example Sentence: John worked for Amazon as a software engineer since 5 years.

4. SRL: Here we would find Semantic Role information by applying SRL
for the verbs having a sense of "work" or "become".

Extracted Template: WORK("John", "Amazon", "software engineer", "") Example Sentence: Amazon's co-founder Jeff Bezos announced about their acquisition of Whole Food Market.

5. Phrase Matching: In this technique, we match phrases based on custom rules like If we have Entity pattern as "ORG" "TITLE" "PERSON", we extract the work template.

Extracted Template: WORK("Jeff Bezos", "Amazon", "co-founder", "") For Template3(PART):

Example Sentence: Dallas is a technical hub in Texas.

1. Lemmatization: We find lemmas for all the tokens in the sentence similarly as we did for Template1 and Template2.
2. NER: We would similarly find NER as we did in Template1 and Template2.
3. Wordnet Features: First of all, we will only trigger this template if we have at least two entities with label "LOC" or "GPE" or "NORP" or "FAC" (using NER). Now we have to find the relation as Loc1 is a part of Loc2. So, we apply various rules on its Wordnet features such as:

a. Check if Holonyms(Loc1) Synonyms(Loc2) NULL
b. Check if Synonyms(Loc1) Meronyms(Loc2) NULL
c. Check if Holonyms(Loc1) Meronyms(Loc2) NULL

Extracted Template: PART("Dallas", "Texas")

C. Full Implementation Details:

a. Programming Tools

To implement this system, we used python3 as Programming language along with various NLP libraries like spacy, AllenNlp and nltk. b. Architecture

Below Image represents the Architecture of the NLP Pipeline developed for Template Extraction.

- Input Document:

o Here, Wikipedia Articles (Unstructured Text) is provided as input to the pipeline.

- Co-reference Resolution:

o Here, the input document is parsed through AllenNlp Coreference Resolution for resolving pronouns with entities.

- Sentencizer:

o After co-reference resolution, it is passed into spacy Sentencizer to segment into sentences.

- Addition of Job Title Entities to Spacy Pipeline: o Spacy doesn't provide Job Title Entity in NER model. So, we provided available job titles from open source database and added to the spacy pipeline.

• Tokenization, Lemmatization, Part of Speech Tagging:

o On each sentence Tokenization, Lemmatization and Part of Speech Tagging is performed to store as features.

o Tokenization: Sentence is tokenized into tokens using spacy. Eg: "Amazon bought Audible."

Tokens: ["Amazon", "bought", "Audible", "."]

o Lemmatization: Each token's lemma is stored as feature. Eg: "bought" – "buy"

o POS Tagging: Each token's associated POS Tag is extracted. Eg: Tokens: ["Amazon", "bought", "Audible", "."]

POS Tags: ["PROPN", "VERB", "PROPN", "PUNCT"]

• Named Entity Recognition (NER), Merge Entities:

o Named Entity Recognition is used to find features like Person, Organization, Location, Geographical Location Entities, Date, Monetary Entity, etc from sentence.

Eg: "Amazon bought Whole Food Market for USD13 Million Dollars." The NER is shown below.

o Merge Entities: Each entities are merged into single tokens for easier referencing to the entities.

• Check Template Triggering:

o Each sentences is checked on triggering of BUY, WORK and PART Template based on various heuristics to achieve efficiency. Once triggered, it performs extraction job.

• Semantic Role Labeling:

o In natural language processing, semantic role labeling is the process that assigns labels to words or phrases in a sentence that indicate their semantic role in the sentence, such as that of an agent, goal, or result.

o The major thematic roles that can be extracted are Agent, Experiencer, Theme, Patient, Instrument, Beneficiary, etc. • Dependency Parsing:

o Dependency Parsing is very useful technique to understand the grammatical structure of the sentence and defines relations between head words and words which modifies other words.

• Phrase Matching:

o Sometimes, template follows certain structure which can be extracted by specifying custom rules for phrases. Spacy provides Matcher tool where we can specify custom rules for matching the phrases.

• WordNet Feature Extraction:

o We are extracting various wordnet features like Hypernyms, Hyponyms, Holonyms, Meronyms, Entailments, etc. to use as features in our template extraction.

IV. Result and Analysis

There were various errors that we encountered while analyzing the result generated by our code. They are as follows:

• Sometimes Spacy's NER doesn't correctly label entities which results in error as we identify entities and use them for template filling in our strategies.

Example Sentence 1: "Richardson is a principal city in Dallas." Spacy's NER annotation:

Instead of detecting Richardson as "LOC", it detected it as "ORG" and so we could not extract PART template from it.

Example Sentence 2: "Amazon sold Whole Food Market to Google." Spacy's NER annotation:

It did not detect Whole Food Market as an "ORG" and so it would not extract the BUY template.

• Sometimes the Coreference resolution also fails to resolve the references correctly. And this cause incorrect extraction of information.

Here, instead of resolving pronoun "he" with "Jobs", it resolved the coreference with Scott McNealy. • Lack of words and its relation in Wordnet Vocabulary. Due to that we cannot find Wordnet features which hinders our extraction of PART template.

Example Sentence: "Richardson is a principal city in Dallas and Collin counties in the U.S. state of Texas." Spacy's NER annotation:

Here Collin is detected as a "GPE"(GeoPolitical Entity), but when we try to find Wordnet relations for Collin, we get nothing.

And so we cannot apply our proposed solution to extract PART template.

V. Summary of Problems encountered and their solution

We faced few problems related to NER for example, spacy's v2.2.0 model detect entity "Souq.com" as "ORG" but spacy's v2.2.5 model doesn't. Same ways entity "Ring" was detected as "ORG" in v2.2.5 but not in v2.2.0. Thus, we decide to ensemble both models to get maximum coverage of detecting entities.

• Also, spacy's NER doesn't provide labelling to Person's Job Title entities. So, we used open-source data for Job Title and then provided it as an input to spacy's EntityRuler which after applying default NER matches patterns provided to it and labels matched entities as "TITLE".

• There were many 3rd party NLP tools that are available for python. And the problem was some provide better results for certain tasks than the other. So, after detailed analysis we concluded to use AllenNlp for Semantic Role Labelling and Coreference Resolution; and Spacy for rest of the tasks.

• For the WORK Template extraction, spacy's NER tagged certain Organization as "PERSON" and so we used both "ORG" as well as "PERSON" label to trigger the WORK Template extraction.

The given solution popped some False Positive which we resolve using dependency parsing.

VI. Potential Improvement

Instead of using default model for NER, SRL and Coreference provided by Spacy and AllenNlp, on top of it, we could train a model to increase our performance.

• We could further find instances where our proposed solution fails and provide a workaround for that.

VII. Reference

1)https://github.com/prit2596/NLP-Template-Extraction/blob/master/Project$_{R}eport_{P}IADA.pdf$
2)$Editorial : MiningScientificPapers : NLP - enhancedBibliometrics$
3)$BagofTricksforEfficientTextClassification$
$Author–ArmandJoulin, EdouardGrave,$
$PiotrBojanowski, TomasMikolov$