5W EXTRACTOR & PREDICATE DISAMBIGUATION

AN EASY INTRODUCTION TO A FAKE NEWS FILTER

ANTONIO MAURO - 1766342

1 Intro

Semantic role labeling is an interesting approach used in order to identify the main predicate of a certain action and knowing it, establish the main actors involved in a certain sentence. Essentially we are just going to identify first of all the predicates (it my be done due to POS tag), knowing it we are going to check the possible arguments it may have and then we identify with a certain accuracy the roles in our tested sentence.

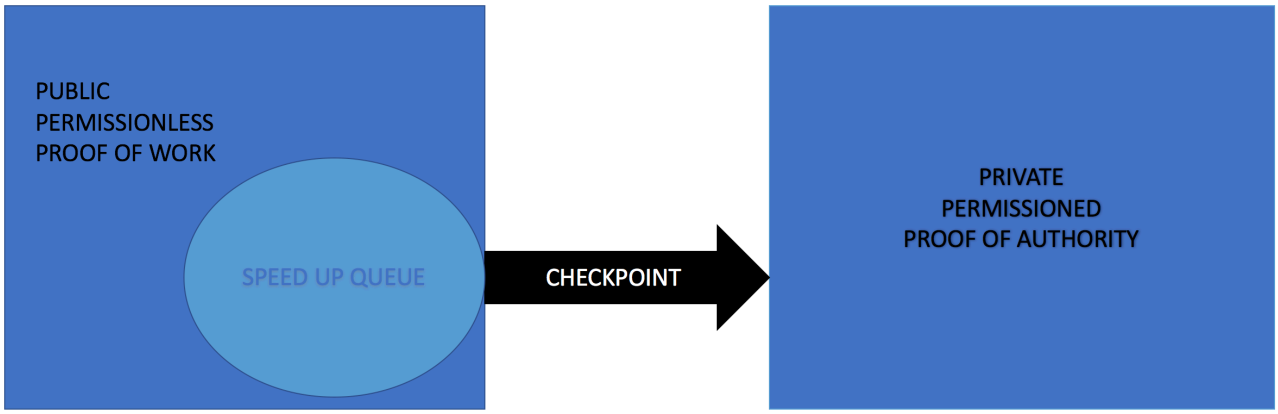
1. Identify and disambiguate predicates
   a. e.g. "He ate an apple" --> "He ate/eat.01 an apple"
2. Identify the arguments of a given predicate
   a. "He ate/eat.01 an apple" --> "He ate/eat.01 an apple"
3. Label the roles of the arguments
   a. "He ate/eat.01 an apple" --> "He_**A0** ate/eat.01 an apple_**A1**"

In terms of main operations, it seems quite easy but each part has to be further detailed in order to be understood better. This approach suggests a way for extracting the relevant aspect on sentences to be analyzed and as a matter of fact it gives the idea for classifying articles. In particular, we are going to discuss the first step (Identify and Disambiguate predicates) used as an add-on for a model consisting in classifying news (so in finding the fake ones)

2 Area of Interest

This improvement may be used In many environments such as information retrieval but, concerning with my experience it may be used in order to classify news. One of the main problem we have to face nowadays is the so called “Fake News”, and one approach may involve blockchain architecture, as a matter of fact saving the whole news causes an easy starvation of the whole system and an impossibility to extract what was acknowledged and saved in order to check if something published is fake. One approach for doing this kind of stuff is concerning with 5w (even many) of an article. An other (more easy but more inefficient) could involve the use of the key-words or entity extracted but as a matter of fact fake news talk about known entities but in a fake context so, it would be a wrong approach, considering only that.

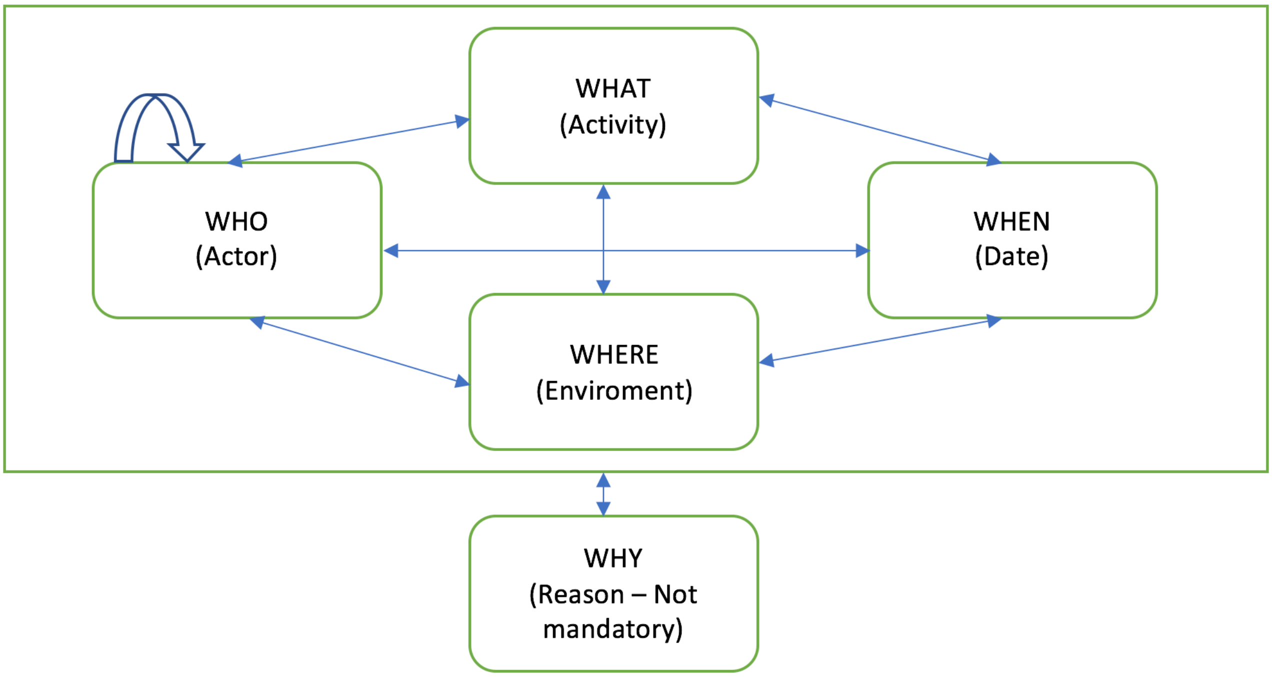
2.1 Architecture - state of art



The blockchain architecture essentially consists of 2 main blocks, the first one, public permissionless, is just like a filter, where we decide if we may let pass a news in the heavy validation phase; the second one represents the last layer of our blockchain architecture, private and permissioned, where we perform the second and last step for the validation of a news.
The decisions about the choice of a first public layer can be resumed in the following way:
In a public layer, everyone can post a news, otherwise we realize a kind of monopoly of information. The structure of meta-info fits better in a public approach so there is no need of using a public-key cryptography. Every news has the same initial score of another, we cannot hack an account and publish whatever we want. In addition we have to remember that presenting a 3-layer architecture, the validity of the news is evaluated better in the last layer, where we have a permissioned blockchain, we just need a ground of trustiness according to the content.

2.2 Meta Information

This part involves NLP algorithm in order to extract the 5w and to disambiguate predicates.
This approach is well used not only to well describe the resource published (it is commonly used in journalism) but being well explicative it may be used to aggregate information that needed to be extracted in the validity phase and fits not only for text but even for images:



Those five main blocks (or four being one optional) well describe every resource. Every one is well explicative but 2 main things has to be pointed out:

- For each resource, there can be multiples sentences that describe it (example an article may describe different aspects)
- WHO is both SUBJECT and OBJECT (notice the arrow). Being it useful when we have to retrieve information we split sentences (maybe in replicated manner):
    - Alice met Bob in Italy
    - Bob met Alice in Italy

  However, we can avoid of splitting sentences (due to avoid replication), we can use an optional WHO as DATIVE, or better the one being affected by the state or action.

Given a sample sentence such as the previous one, we may understand that a factor useful to describe the 5W is not only the main content but even the accuracy of information extracted. Summing up the main things said until now we may describe (in a JSON like format) the metadata:

```
{
    name: "example.pdf",
    hash: "0x0123abc",
    included-resources: ["hashOf1", "hashOf2"],
    content: {[
        who: {
            name: "Alice",
            accuracy: 0.7,

        },
        what: {
            name: "meet.01",
            accuracy: 0.7,

        },
        dative: {
            name: "Bob",
            accuracy: 0.7,

        },
        where: {
            name: "Italy",
            accuracy: 0.2,
```

```
        },
        when: {
            name: null,
            accuracy: 0,

        },
        why: null], ...
    }
}
```

The first time, so we want to accept the news but be suspicious with reference to its correctness. So another field "trustiness" has to be added with a value between 0 and 1. This may give an hint on consensus algorithm to be adopted which involves to accept or reject news according a certain threshold.

2.3 Evaluate 5W

Essentially before posting we trigger an API call to retrieve the 5w of the article (black box part), we evaluate the hash of the resource, so we spread to everyone the meta-info of the hashed resource with 5w extracted all linked with the original resource.
In particular the meta-info structure prepared in this phase has all the field filled with the only exception of the "trustiness" because this value will be evaluated only at the end of the first block.
Being this phase modeled as black box, we have to discuss which API can we use, in particular we are interested in the following project

- https://github.com/fhamborg/Giveme5W.git
    - This is an open source project concerning with the extraction of the 5W in a text
    - Available tutorial on main site to make it work, it works perfectly with Linux

Even if it works only in English and does not have disambiguation and dative element it is the good starting point, the rest can be easily integrated.

2.4 Check validity 5w

For this step is proposed an algorithm that analyze the like hood that a certain event may happen regarding the 5w list collected at the beginning.
This step is the most important and it gives as output a numeric value between 0 and 1 and according to this value we may decide if accept or reject a news. The main operation that should be performed are:

- Retrieve the meta-info of linked resources.
- Check the semantic compatibility of the 5w model of those linked resource with the one to be uploaded.

Remembering that a resource file can be described by many 5w sentences, discarding the ones whose ground of trustiness has already been checked due to the ground of trustiness of the claimed resources we have to assure the validity of the other ones, if present :

- Check the semantic compatibility between the already checked sentences and the other ones.
- Check the semantic compatibility between each other of all the sentences.
- For each sentence check the validity consulting the ledger.

In case that there are no other sentences to check, so all of them have a claimed source, we may trigger the spread of this news to the layer 3. We will focus on later.

We can imagine a validity check in the following way. Fixed the 5w we can evaluate the probability that the sentence is true or better compatible with the rest.
The why option can be discarded because is not useful in this case (WHY can be used in a more semantic detailed analysis).
Let us consider two who A, B (B is the facultative dative one), when D, where E and what C, we have to estimate the following probability:

$$P(\mathrm{ABCDE}) = \mathrm{P(A)P(B|A)}\mathbf{P(E|AB)P(D|ABE)P(C|ABDE)}$$

The first two can be not considered but evaluating the third and fourth we can say that WHERE and WHEN are useful only if are considered together (they give a kind of context) and the last one (the action) is useful only regarding the probability that a person performs some action, but given an estimation of the usual context of the main actor involved and the probability that a certain action may happen given WHERE and WHEN we can redesign the probability in the following way:

$$P(Sentence\ is\ true) = P(AB|Where, When)P(C|A, B, Where, When) =$$
$$= P(A|Where, When)P(B|Where, When)$$
$$P(A\ did\ a\ certain\ Action)P(B\ dir\ OR\ received\ a\ certain\ action)P(Action|Where, When)$$

Doing the same calculus starting from Where and When joined as D we have:

$$P(Sentence\ is\ true) = P(ABCD) = P(D)P(C|D)P(AB|CD) =$$
$$= P(D)P(C|D)P(A|CD)P(B|CD) =$$

… we do the calculus for A but it is the same for B …

$$= P(D)P(C|D)\frac{P(A)}{P(CD)}P(CD|A)P(B|CD) =$$
$$= P(D)P(C|D)\frac{P(A)}{P(CD)}P(C|A)P(D|A)P(B|CD) =$$
$$= P(D)P(C|D)\frac{\cancel{P(A)}}{P(CD)}\frac{P(D)}{\cancel{P(A)}}P(A|D)P(C|A)P(B|CD)$$

… being $\frac{P(D)}{P(CD)} \cong 1$ … so being negligible such as P(D) …

$$P(Sentence\ is\ true) = P(C|D)P(A|D)P(C|A)P(B|D)P(C|B)$$

Now we can imagine a kind of algorithm (similar to **simulated annealing** but in the reverse manner) that gives as output the first estimation of trustiness.

1. Fix a value T (the minimum validity of news we may collect, so we start by the most valid so not new) and estimate value N (Number of Doc useful remembering the total number of Doc analyzed) that is correlated to T
2. Fix the actor A and find all Where possible according to accuracy
3. Fix the actor A and find When given Where according to accuracy
4. Calculate the first probability, if the number of Doc analyzed is greater than N, stop else go deeper
5. Perform step 2,3,4 for actor B (dative) if present
6. Given the context evaluate the last 3 probabilities (like previous)
7. Multiply the total result with T and compare with a threshold to decide if accept or not a news

What is important to be remembered is: If performing step 2,3 we find something not compatible we have as outcome 0 (all results are multiplied), otherwise a value to be compared with threshold.

```
T = 0.8;
N = 10/T;
critical = minimum_ground of trustiness

while (T > critical) {
    accuracy = 0.8
    while (accuracy > 0) {
        listWhere = list_all Where by trace according accuracy
        listContext = [];
        for(where in listWhere)
            listContext.add(Fix the actor A and find When given Where above accuracy)
        evaluate probability update GLOBALLY
        if(listContext.size > N) break
        else
            accuracy -= 0.2
    }
    T -= 0.1
    N = 10/T
}
```

Step 6 is based upon a similar algorithm, we proceed according to T and N, so we find the occurrences (action given A, given B, given Where and When) over the total number of sentences retrieved to estimate the last 3 probabilities.
As a matter of fact at the end we simply multiply the five probabilities, but as a more practical result (event if not strictly mathematical rigorous) we can estimate the average. In fact given two values such as 0.8 and 0.9 it is better to save the average, so 0.85 instead of multiplying, because we have an high score to compare with threshold, otherwise we should decrease the threshold.

2.5 Revisited Mining

The most innovative concept behind this architecture at the last layer is how is considered mining. First, we should say that the whole block is considered as a black box one, we are going to consider the main ideas behind it, so we are going to define how consensus is going to be achieved and how witnesses are going to be evaluated, but we are not going to define the way in which a news is going to be checked.
Each node can even be composed by physical people that downloads the article and manually revise is. Otherwise a certain trusted company can even propose an own algorithm and use this one to validate the news. The great idea is that we present a private blockchain with trusted entities but we do not force them to use a certain validity check method, because at the end they will promote an article and so their name will be their sort of revenue.
Miners do not earn on transaction, like cryptocurrencies, their main purpose is that their name will be an output for the news. In certain sense, they may bet on the validity of a news and use their name as guarantee. Afterwards the news with the name will be publicly available and they can monetize as they usually do.
Obviously, they can even be the promoter of an article, so we can refine this last layer not only to check the validity of anonymous news, but even for checking news published by one of the trusted entities. Considering the first layer public and permissionless it means that in this case this article can only be present at this level, as a matter of fact the amount of witnesses is static, maybe can be dynamic but in this case proportional to the source. However, this last point can be seen as a future work, for this moment it will not be deeply analyzed.

3 Predicate Identification and Disambiguation

In order to perform this first part two solutions are presented but both have a first main part that involves predicate identification. Predicate identification can be done in few lines of code importing nltk library in order to perform POS tagging:

- We tokenize our sentence. It can be even done without using any library, it is just a split of a string saving its elements in an array.
- Using this tokenized sentence as input we may use **nltk.pos_tag** to tag with POS element our token.
- Now we have an array where we have to check the verb element and it is an easy operation.

Identified the predicate, we have to disambiguate it knowing the context in which it is used. The first and easy to integrate approach involves nltk library and lesk algorithm. This kind of approach can be considered as a knowledge based one, in fact lesk algorithm is based upon overlapping of context and gives as output the most frequent and so suitable sense. Using it in with nltk standard implementation, in order to compute overlapping it reads a data file data.pos that in our case is data.verb. This function gives as output the verb sense or None so, in this last case, we can retrieve the first sense using other resources such as BabelNet or we may decide to save an UNK sense or simply the verb lemma without any sense. The system cannot be trained, we have about 10% of sense matching with CONLL2009-TRAIN, but the knowledge base can be populated with new sense with the care of adding them in the right place with the right structure in data.verb file.
The second approach can be of an easy LSTM, where we train our system sending as input the verb identified and the context sequence, expecting the right label (so sense) for the verb. We have to identify a vector representation for our input, maybe using genism library and so Word2Vec approach. In this case the accuracy is going to be higher, otherwise the system needs to be trained and so it is a more time consuming operation starting from 0.

3.1 Lesk Algorithm

Deciding to follow the first approach, we will describe the main step of the lesk algorithm. It is based on the assumption that words in a given "neighborhood" (section of text) will tend to share a common topic.
For every sense of the word being disambiguated one should count the amount of words that are in both neighborhood of that word and in the dictionary definition of that sense then the sense that is to be chosen is the sense which has the biggest number of this count

```
function SIMPLIFIED LESK(word,sentence) returns best sense of word
    best-sense <- most frequent sense for word
    max-overlap <- 0
    context <- set of words in sentence
    for each sense in senses of word do
        signature <- set of words in the gloss and examples of sense
        overlap <- COMPUTEOVERLAP (signature,context)
        if overlap > max-overlap then
                max-overlap <- overlap
                best-sense <- sense
end return (best-sense)
```

Unfortunately, Lesk's approach is very sensitive to the exact wording of definitions, so the absence of a certain word can radically change the results. As a matter of fact dealing with news that with high frequency talk about known actors and actions it could be a good starting point.

4 APPENDIX

4.1 PREDICATE DISAMBIGUATION SCORE

After testing the matching of the disambiguated verbs on CONLL2009 training and development data, we realized that we do not have a perfect alignment of disambiguated verb name:

```
Let  us start !!!
Total verbs = 179014
Correct matches = 17620
Nones = 44582
Number of sens = 26277
Score 9.842805590624197%
```

```
Let  us start !!!
Total verbs = 6390
Correct matches = 629
Nones = 1543
Number of sens = 930
Score 9.843505477308295%
```

Let us define the total number of verb identified, then the correct matches (so the score dividing both results) and then to better evaluate the system we found the points of improvements:

- Nones result : result may have an higher score if we enrich the corpus
- Number of sens: Defining that there is not a perfect alignment but often even if the name of the disambiguated verb is not the same, the synset number for that verb is the same.

With a more precise alignment it would be possible to reach up to 20% of score, but integrating the nones enriching the corpus we may have up to 40%. As a matter of fact using test dataset we labeled 94% of identified verbs.

4.2 ARCHITECTURE'S OUTPUT

Summing up all considerations, the system concerning with 5w extraction and data preprocessing works in the following way:

- With Giveme5w project we extract the main elements.
- Being this output in the for {who:[],where:[]…} we adapt it as a list of five W sentences.
- Upon performing this rewrite operation, we preprocess the what element using lesk algorithm and we save in this field his sense or if not present just the verb lemma.
- The last element to identify is the dative one, essentially we evaluate the POS tags of the what sentence extracted finding the object that is not in where or when set.


```
{
    "whatName": "hat.02",
    "whatScore": 50,
    "whenName": "Friday  - ",
    "whenScore": 16.0,
    "whereName": "Naples ",
    "whereScore": 50,
    "whoName": "Carl ",
    "whoScore": 50
}
```

```
{
    "dativeName": "Bob",
    "dativeScore": 50,
    "whatName": "sleep_together.01",
    "whatScore": 50,
    "whenName": "Friday  - ",
    "whenScore": 16.0,
    "whoName": "Alice ",
    "whoScore": 50
}
```


As a matter of fact metadata can be improved, such as the value of accuracy of every element. For now we retrieve the accuracy value of the when element from the API involved in extraction the rest is set to a default value of 50%. Future works may involve the use of BabelNet API to disambiguate When element (setting the accuracy value involved in geolocalization) and doing a similar job for the accuracy of What and Dative concerning with entity extraction, but this last phase is more difficult to express.

4.3 Smart Contract

As presented in section 2.4, using probability we have defined the threshold algorithm written in solidity language, where we are really constrained even in terms of variables that can be used:

```
function computeTrustiness(string who, string where, string when, string what, string dative) internal view returns(uint) { //other
    uint T = 8000; //treshold for trustiness uint N = T/10; //max num of document to be analyzed
    uint critical = 2000; //critical value (minimum for trusted news)
    uint accuracy = 8000; //defined as trustiness of document

    while (T > critical) {

        while (accuracy > 1) {
            //compute P(A|Where&When)*P(B|Where&When)*P(Action|Where&When)*P(Action|A)*P(Action|B)
            //LET US DEFINE AN ARRAY OF UINT OTHERWISE STACK TOO DEEP (MAX 16)
            //error DIVISION BY ZERO
            uint[] memory probCount = new uint[](8);
            probCount[0] = 0; //countContext
            probCount[1] = 0; //countActorinContext
            probCount[2] = 0; //countDativeinContext
            probCount[3] = 0; //countActioninContext
            probCount[4] = 0; //countActor
            probCount[5] = 0; //countDative
            probCount[6] = 0; //countActioninActor
            probCount[7] = 0; //countActioninDative
            for (uint i = 0; i < questions.length; i++){
                if (compareStrings(questions[i].whereName,where) &&
                        compareStrings(questions[i].whenName,when) &&
                        questions[i].whereAccuracy >= accuracy &&
                        questions[i].whenAccuracy >= accuracy) {
                    probCount[0] += 1;
                    if (compareStrings(questions[i].whoName,who)) {
                        probCount[1] = probCount[1] + questions[i].whoAccuracy;
                    }
                    if (compareStrings(questions[i].dativeName,dative)) {
                        probCount[2] = probCount[2] + questions[i].dativeAccuracy;
                    }
                    if (compareStrings(questions[i].whatName,what)) {
                        probCount[3] = probCount[3] + questions[i].whatAccuracy;
                    }
                }
                if (compareStrings(questions[i].whoName,who) &&
                        questions[i].whoAccuracy >= accuracy) {
                    probCount[4] += 1;
                    if (compareStrings(questions[i].whatName,what)) {
                        probCount[6] = probCount[6] + questions[i].whatAccuracy;
                    }
                }
                if (compareStrings(questions[i].dativeName,dative) &&
                        questions[i].dativeAccuracy >= accuracy) {
                    probCount[5] += 1;
                    if (compareStrings(questions[i].whatName,what)) {
                        probCount[7] = probCount[7] + questions[i].whatAccuracy;
                    }
                }
            }
```

5 REFERENCES

- Huckle S, White M (2017) Fake news: a technological approach to proving the origins of content, using blockchains. Big Data 5:4, 356–371, DOI: 10.1089/big.2017.0071.
- Luheng He and Kenton Lee and Mike Lewis and Luke S. Zettlemoyer (2017) Deep Semantic Role Labeling: What Works and What's Next. ACL
- Barriocanal E, Alonso S, Sicilia M, Deploying Metadata on Blockchain Technologies. V.755, 2017, Pages 38-49 11th International Conference on Metadata and Semantic Research, MTSR 2017
- Juan Benet. IPFS—Content Addressed, Versioned, P2P File System (DRAFT 3). 2017.
- Kim, Henry & Laskowski, Marek. (2016). Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance.
- Das, Amitava & Ghosh, Aniruddha & Bandyopadhyay, Sivaji. (2010). Semantic role labeling for Bengali using 5Ws. 1 - 8. 10.1109/NLPKE.2010.5587772.
- Aniello, Leonardo & Baldoni, Roberto & Gaetani, Edoardo & Lombardi, Federico & Margheri, Andrea & Sassone, V. (2017). A Prototype Evaluation of a Tamper-Resistant High Performance Blockchain-Based Transaction Log for a Distributed Database. 10.1109/EDCC.2017.31.
- Buterin, Vitalik & Griffith, Virgil. (2017). Casper the Friendly Finality Gadget.

Interesting web sites:

- https://dzone.com/articles/blockchain-and-ai-can-defeat-fake-news
- https://github.com/glowkeeper/Provenator
- https://github.com/professormarek/traceability
- https://dandelion.eu/semantic-text/
- https://github.com/fhamborg/Giveme5W.git
- https://github.com/jgung/semantic-role-labeling
- https://www.nltk.org/
- https://en.wikipedia.org/wiki/Lesk_algorithm