Matching Ukrainian Wikipedia Red Links with English Wikipedia's Articles

Kateryna Liubonko
Ukrainian Catholic University
Lviv, Ukraine
aloshkina@ucu.edu.ua

Diego Sáez-Trumper
Wikimedia Foundation
San Francisco, USA
diego@wikimedia.org

ABSTRACT

This work tackles the problem of matching Wikipedia red links with existing articles. Links in Wikipedia pages are considered red when lead to nonexistent articles. In other Wikipedia editions could exist articles that correspond to such red links. In our work, we propose a way to match red links in one Wikipedia edition to existent pages in another edition. We define the task as a Named Entity Linking problem because red link titles are mostly named entities. We solve it in a context of Ukrainian red links and English existing pages. We created a dataset of 3171 most frequent Ukrainian red links and a dataset of almost 3 million pairs of red links and the most probable candidates for the correspondent pages in English Wikipedia. This dataset is publicly released[1]. In this work we define conceptual characteristics of the data — word and graph properties — based on its analysis and exploit these properties in entity resolution. BabelNet knowledge base was applied to this task and was regarded as a baseline for our approach ($F1$ score = 32 %). To improve the result we introduced several similarity metrics based on mentioned red links characteristics. Combined in a linear model they resulted in $F1$ score = 85 %. To the best of our knowledge, we are the first to state the problem and propose a solution for red links in Ukrainian Wikipedia edition.

CCS CONCEPTS

• **Information systems → Information extraction**.

KEYWORDS

Wikipedia, Entity linking, Red links

ACM Reference Format:
Kateryna Liubonko and Diego Sáez-Trumper. 2020. Matching Ukrainian Wikipedia Red Links with English Wikipedia's Articles. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3366424.3383571

1 INTRODUCTION

Nowadays Wikipedia is constantly attracting attention of Data Scientists and Machine learning engineers. First, this multilingual encyclopedia is a subject of study per se. Secondly, it is used as a knowledge base to develop other tools (e.g. DBpedia[2], BabelNet[3]) and training Natural Language Processing (NLP) algorithms[1]. The proposed work addresses both ideas. It tackles the problem of gaps in Wikipedia network to remove them and at the same time exploits Wikipedia as means to do it.

This work refers to Wikipedia gaps caused by so-called red links. The phenomenon of red links in Wikipedia has not been studied deeply yet. Red links are links to pages which do not exist (either not yet created or have been deleted). The problem of red links is that they can refer to Wikidata items or Wikipedia articles which already exist in other languages but can not been identified from the source language. For example an article in language L_i can contain a red link to an article about A_i which does not exist in L_i but exists in another language L_j. Our goal is to identify such connections between missing content in one language with existing content in another language. We tackle this problem in a context of Ukrainian and English Wikipedia editions. Our solution is developed on red links of Ukrainian Wikipedia edition looking for the correspondence on the English Wikipedia edition.

The number of red links in Ukrainian Wikipedia is 1 554 986[4] unique titles while the size of Ukrainian Wikipedia itself is 817 892 existent articles. In its turn, English Wikipedia is 8 times bigger than the Ukrainian edition (5 719 743 articles). Therefore, the idea is to use the English version as a knowledge base to fill the gaps of Ukrainian red links.

Several projects in English Wikipedia community were held to tackle the problem of matching existing red links with existing items. Some of the most relevant for our work are Red Link Recovery Wiki Project[5] and Filling red links with Wikidata project[6]. Still, they are either only discussed as an idea and not implemented or not currently maintained.

The alternative to red links in Wikipedia can be considered templates with interlanguage links. They refer to nonexistent articles as well as red links but also link to existent articles in another languages, for example in English edition. Thus, they are a better alternative for red links and red links could be transformed into such templates.

We approach the problem of red links as a Named Entity Linking task [14]. The reason is that the majority of Wikipedia articles are about Named Entities, and we solve these entities (red links) linking them to English articles. We use graph and word properties of Wikipedia articles and apply different similarity metrics to find the correspondent items in English Wikipedia for Ukrainian red

[1] https://doi.org/10.6084/m9.figshare.11550774

[2] https://wiki.dbpedia.org/

[3] https://babelnet.org/

[4] Wikipedia dumps from the 20th of September 2018

[5] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Red_Link_Recovery

[6] https://meta.wikimedia.org/wiki/Filling_red_links_with_Wikidata

WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3383571

links. Finally, we compare the results of these metrics with the results of BabelNet knowledge base considering the last one as our baseline. Among all the applied techniques we present the similarity model that produces the best results.

We consider the following to be the main contributions of our work:

- We present a solution for filling the gaps in Ukrainian Wikipedia network using the English Wikipedia edition as a knowledge base. To the best of our knowledge, this is the first work tackling the problem of red links in Ukrainian Wikipedia.
- We create a dataset of 2 957 927 pairs[7] of red links and the candidate articles in English Wikipedia for the most frequent 3 171 red links from Ukrainian Wikipedia. We manually labeled them and publicly release it.
- We present a data analysis of red links in Ukrainian Wikipedia which can foster further investigation in this field.
- We publicly release the code on github[8].

2 RELATED WORK

2.1 Wikimedia projects and tools

To the best of our knowledge, there are not scientific publications working on matching red links to Wikidata items. Several projects were held by Wikipedia community but with no publicly published peer-reviewed papers.

The *Red Link Recovery Wiki Project for English Wikipedia* [9] project had been active until 2017. The main goal of this project was to reduce the number of irrelevant red links. Within the project they developed a tool to suggest alternative targets for red links in the same Wikipedia edition. Among the techniques used in a tool were comparing titles by a weighted Levenshtein edit distance, creating names with alternate spellings, matching with titles transliterated (from originally non-Latin entities), using alternative systems (e.g. Pinyin, Wade-Giles), matching with titles spelled with alternative rules (e.g. anti personnel / anti-personnel / antipersonnel). We considered some of these techniques in our work.

Project proposal *Filling red links with Wikidata*[10] was for the first time explicitly described in Wikimedia mailing list in 2014. Its aim was to make red links a part of a Wikipedia graph which is similar to ours but is related with the particular moment of creating a red link. The idea was to create placeholder articles filled with data from Wikidata. This project proposal has a wide perspective not only connecting red links to Wikidata items but also automatically creating Wikipedia pages. However, it was not implemented. The discussion on that project involved many questions on how to maintain and edit these new 'semi-articles'.

Also, the suggestion to connect red links to Wikidata items appeared in Wiki-research-l Digest, Vol 157, Issue 19[11]. But many issues arose that were related to the process of connecting red links to the appropriate Wikidata items while creating them. The project was not implemented.

The work described above is all in the domain of English Wikipedia edition. For Ukrainian edition the only thing that was found related to the red links problem is gathering lists of red links and combining them into topics.

2.2 BabelNet

BabelNet[12] is a knowledge base that serves as a multilingual encyclopedic dictionary and a semantic network. It is initially constructed on Wikipedia concepts and WordNet[13] database. The main idea behind BabelNet is that encoding knowledge in a structured way helps to solve different NLP tasks even better than statistical techniques.

BabelNet had been applied to Entity Linking tasks before and showed good results[9] for finding the correct translations for multimeaningful words. It worked especially well for nouns and noun phrases. The majority of Wikipedia titles are of these categories.

Thus, we suggested BabelNet a tool that can work out for our problem.

2.3 Entity linking

Named Entity Linking is a task to map a named entity mentioned in a text to a corresponding entry stored in a existing knowledge base [14]. Entity linking serves for information retrieval tasks such as creating text summary, search engines, also helps with augmenting text with links and so on.

Resources used for Entity Linking tasks are mainly Wikipedia and its subsequent projects such as DBpedia, Wikidata[14], YAGO[15] [4], Freebase [3].

The theoretical basis for Entity linking pipeline we derived from the work [11] as they present a good overview of main approaches to entity linking. According to the authors, Entity Linking pipeline typically consists of the following modules:

1. Candidate Entity Generation: for each entity mention m ∈ M filter out irrelevant entities in the knowledge base and retrieve a candidate entity set E_m which contains possible entities that entity mention m may refer to.
2. Candidate Entity Ranking: leverage different kinds of evidence to rank the candidate entities in E_m and try to find the entity e ∈ E_m which is the most likely link for mention m.
3. Unlinkable Mention Prediction (optional): validate whether the top-ranked entity identified in the Candidate Entity Ranking module is the target entity for mention m.

This work [11] shares the basic ideas on how to approach the Entity Linking task in the context of Wikipedia, and we apply it to our problem of linking red links to English Wikipedia.

2.4 Embeddings for concept similarity

The concept of embedding has roots in topology, differential geometry and category theory [8]. In these fields 'embedding' means mapping from domain X into co-domain Y $f : X \to Y$. This map

[7] Wikipedia dumps from the 20th of September 2018

[8] https://github.com/Katerali/Red_links_Project_for_Wiki

[9] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Red_Link_Recovery/RLRL

[10] https://meta.wikimedia.org/wiki/Filling_red_links_with_Wikidata

[11] https://lists.wikimedia.org/pipermail/wiki-research-l/2018-September/006439.html

[12] https://babelnet.org/

[13] https://wordnet.princeton.edu/

[14] https://www.wikidata.org/wiki/Wikidata:Main_Page

[15] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

is injective (each $y \in Y$ has only one corresponding $x \in X$) and structure preserving. The kind of structure preserved in mapping depends on X and Y.

Among considered works on this topic we derived our ideas mostly from [12] and [15]. In the first, authors applied embeddings for a Named Entity Linking task proposing an effective graph-based algorithm which exploits embeddings on two levels, a word level and a document level. In this way they captured the meaning of a word and a meaning of the context the word is used in (e.g. topic). In the second, embeddings were applied to Concept Analogy and Concept Similarity tasks. Here each Wikipedia article was embedded as a separate concept.

Concerning our particular task, however, in the area of graph embeddings we have not found applications of such techniques for entity resolution in Wikipedia nor Wikidata.

In its turn, in the field of Entity linking task there is a developed methodology both in general and in the context of Wikipedia.

3 SIMILARITY-BASED MODEL FOR ENTITY LINKING TASK

In this section we present our solution for red links matching problem. We introduce two macro-approaches which help to catch different features of the data. The first is based on graph properties and the second is based on word properties. Then we combine them using a linear model.

3.1 Similarity based on Graph properties

Representing Wikipedia as a graph we refer to its elements as nodes and links between nodes. We see the Wikipedia graph in the following way:

- Nodes of the Wikipedia graph are existent articles and red links;
- Incoming link with regard to an article is a link to this article. In Wikipedia, it means that the considered article is mentioned in another article;
- Outgoing link with regard to an article is a link from this article. In Wikipedia it means that the considered article mentions another article;
- Concurrent links are outgoing links from the same article. In Wikipedia, it means the considered that articles occur in the same Wikipedia article.

We can represent it as in the figure 1.

In figure 1 A), F_1, F_2 and F_3 are existent Wikipedia articles. R is a red link. The specifics of this graph is that all links are directed. Nodes which represent existent articles can have both incoming and outgoing links as nodes F_1 and F_2.

Red links articles do not have outgoing links. They can be described by incoming and concurrent links. In figure 1 B) the incoming links for a red link R are from nodes F_2 and F_3.

The concurrent link for the red link R in this case is the edge from node F_2 to node F_1 (figure 1 C)).

3.1.1 Calculating links in common with Jaccard similarity. As follows from the characteristics of red links, incoming and concurrent links can be used as their features by which we compare them with other Wikipedia articles. The similarity measure that we use is Jaccard score [7]. It is applied to compare sets which are unordered collections of objects. The general idea behind it is to calculate the fraction of common elements in the considered sets over all the elements of these sets. In terms of sets it is defined as an intersection over union and formalized in the following way:

$$S_{AB} = \frac{A \cap B}{A \cup B}, \tag{1}$$

where A and B are two sets to compare.

Jaccard similarity metrics results in a number within the margin from 0 to 1, where 0 means no similarity and 1 means totally similar.

In case of comparing articles by incoming links we are going to represent each article (that is each node) as a set of its incoming links. Then our formula will be the following:

$$S_{I_{eng}I_{ukr}} = \frac{I_{eng} \cap I_{ukr}}{I_{eng} \cup I_{ukr}}, \tag{2}$$

where I_{eng} is a set of incoming links for an English non-translated article and I_{ukr} is a set of incoming links for a Ukrainian red link. Thus we can compare articles by pages where each one occurs.

In case of comparing articles by concurrent links we are going to represent each article as a set of its concurrent links. Then our formula will be the following:

$$S_{C_{eng}C_{ukr}} = \frac{C_{eng} \cap C_{ukr}}{C_{eng} \cup C_{ukr}}, \tag{3}$$

where C_{eng} is a set of concurrent links for an English non-translated article and C_{ukr} is a set of concurrent links for a Ukrainian red link.

Thus, we can compare articles by links which occur in the same page.

3.1.2 Graph embedding. For this problem the theoretical and software background was based on the paper of Palash Goyal and Emilio Ferrara [5] and their python library GEM 5[16].

Among different embedding techniques described in that article and implemented in the library we have chosen Locally Linear Embedding and Structural Deep Network Embedding (SDNE). The choice of Locally Linear Embedding was due to way it embedded the nodes — it assumes that every node is a linear combination of its neighbors in the embedding space. SDNE was chosen due to its good results in experiments provided by authors of the article.

3.2 Similarity based on Word properties

3.2.1 Levenshtein edit distance. Levenshtein distance is one of the best approved metrics to measure the similarity between two sequences of symbols. It is called 'edit distance' because we calculate how many edit operations are needed to transform one string into another. [13] The edit operations are deletion, insertion and substitution.

A formal definition of the Levenshtein distance (introduced by Dan Jurafsky using concepts of dynamic programming [6]) is

[16] https://github.com/palash1992/GEM

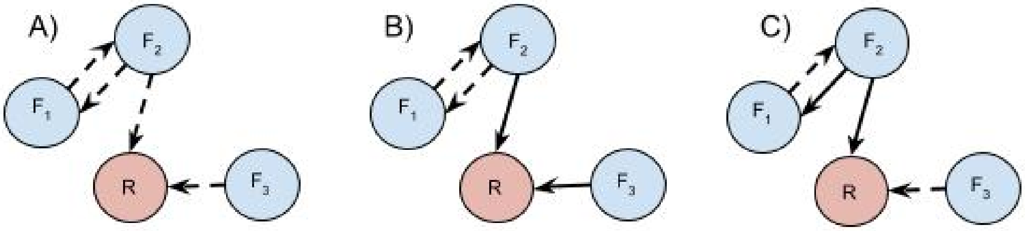

Figure 1: Considered types of connection in Wikipedia graph

A) General representation of a part of a Wikipedia graph; B) Incoming links for a red link; C) Concurrent links for a red link

$$D[i,j] = min \begin{cases} D[i-1,j] + del - cost(source[i]) \\ D[i,j-1] + ins - cost(target[j]) \\ D[i-1,j-1] + sub - cost(source[i], target[j]) \end{cases} \tag{4}$$

Here source[i] is a position of a character in a source string (which we compare) and target[j] is a position of a character in a target string (to which we compare).

There can be slight modifications for this edit measure. First, each operation (insertion, deletion, substitution) can be weighted differently. In our case we leave it with default uniform weights where each operation costs 1. Second, minimum edit distance can be applied in Generalized Levenshtein Distance form or be normalized. Normalization is done because sequences have different sizes and as pointed out in [13], two errors for short strings cost more than for the long ones. So in our project we refer to edit distance normalized by the longest string between a red link and a candidate.

This metrics results in a number within the margin from 0 to 1, where 0 means that items are the same (edit distance is 0) and 1 means that they are totally different.

Levenshtein edit distance compares sequences of a common graphical system (e.g. Latin script). Therefore, applying it to items of different languages requires the transliteration step. Transliteration is a mapping of symbols from one language system to another according to particular linguistic rules. There could be different sets of rules and thus different ways to transliterate from one language to another.

3.2.2 Cross-lingual Word Embedding. Cross-lingual word embedding is a transfer of monolingual word embedding techniques (which is a vector representation of words in a linear space) into the context of several languages. Hence, a notion of a joint embedding space is introduced. Two main reasons of using cross-lingual embeddings are highlighted by [10]. First, they enable us to compare meanings of words across languages, which is a key step in a bilingual lexicon induction, machine translation, or cross-lingual information retrieval, for example. Second, cross-lingual word embeddings enable model transfer between languages, e.g. between resource-rich and low-resource languages, by providing a common representation space.

In our task we are based on the fastText library for learning text representations because it is trained on a Wikipedia corpora. As [10] conclude from their research, data plays the key role in a process of aligning a cross-lingual representation space and is more important than an underlying software architecture. FastText and a Babylon multilingual project[17] are employed to create a tool for mapping word meanings for 78 languages. Vector representations for Ukrainian words are trained as well.

With this tool we appeal to a mapping-based approach of cross-lingual embedding. This method consists in training word embedding separately in different languages and then align them using some dictionary. Then a transformation matrix is searched to switch between spaces. With this matrix cross-lingual tasks are performed.

3.3 Multi-factor Similarity-based Model

Based on defined properties (graph similarity and word similarity) we can apply different models to solve our task. Those properties could be considered as factors (features) and the problem could be formulated using standard machine learning concepts. We will treat the problem as a supervised modeling, where an instance is a collection of obtained properties and a label is a boolean that shows whether a candidate article is the actual correspondent page to a red link or not. With such settings we have a binary classification problem.

Taking into account the fact that there are only four features, we will concentrate on a linear model for binary classification. Linear model is simple and robust and can serve as a starting point for future modeling. Moreover, results are highly interpretable, given a clear notion of features' importance.

Logistic regression is a model for binary classification with linear decision boundary [2]. The model predicts the posterior probability of one class C_1 ('true') based on a feature vector $\boldsymbol{\phi}$:

$$p(C_1|\boldsymbol{\phi}) = \sigma(\mathbf{w}^T\boldsymbol{\phi}) = \sigma(w_0 + w_1 \cdot \phi_1 + ... + w_n \cdot \phi_n), \tag{5}$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the logistic sigmoid function, n is a size of a feature vector (4 in our case), ϕ_i are features themselves, w_i are parameters of the model. The probability of a second class ('false') is obtained by $p(C_2|\boldsymbol{\phi}) = 1 - p(C_1|\boldsymbol{\phi})$.

Based on a maximum likelihood we obtain the procedure of training this model.

[17] https://github.com/babylonhealth/fastText_multilingual

4 DATASET

4.1 Data collection & pre-processing

To the best of our knowledge there is no pre-existing dataset with labeled data to match red links from Ukrainian to English Wikipedia. Thus, we have created it on the basis of Wikipedia XML dumps (September 2018), SQL dumps of links between Ukrainian and English Wikipedia articles and a Wikipedia pages network.

4.1.1 Data retrieval and pre-processing of the whole dataset. Our goal was to obtain red links of Ukrainian Wikipedia edition and all the corresponding information that would help to solve our matching problem. Data retrieval and some parts of pre-processing were done based on the on our previous work[18].

The size of English Wikipedia is 28.0 GB in a compressed format[19]. It contains 5 719 743 full English articles. Whereas Ukrainian dump’s size which we took as an input is 2.1 GB[20]. It contains 817 892 Ukrainian Wikipedia articles of full size. The special approach was required to process this data on one computer. Mostly we split it into chunks and processed them one by one.

We obtained 2 443 148 red links in Ukrainian Wikipedia among which 1 554 986 are unique titles. Among them there are 1 010 955 red links which occur only once. And the most frequent link occurs in 941 articles (’ацетилювання’ (’atsetyljuvannja’)).

To compose a set of candidates in English Wikipedia we retrieved all non-translated English articles, the correspondences between Ukrainian and English articles and all the incoming links to non-translated articles in English Wikipedia. In English Wikipedia the number of articles not translated to Ukrainian is 5 264 607 which means that only 8 % of English Wikipedia is translated into Ukrainian. And vice versa the number of links between Ukrainian and English Wikipedia is 599 636 which is 73 % of all Ukrainian Wikipedia articles.

4.1.2 Retrieving the sample. For the reason that we can’t obtain the ground truth for such amount of red links and due to available hardware resources[21], we decided to work for our project with samples. We obtained a sample of 3 372 red link titles which were in Ukrainian Wikipedia by the 20th of September 2018. The sample was obtained by choosing red link titles that occur in 20 or more articles which have corresponding articles in English Wikipedia.

4.1.3 Characteristics of the obtained sample: 96 % of the sample are Proper Names. They include names of people, animal species (mostly moths), plant species, sport events, names of publishing houses, media sources, geographic locations and territories (mostly French regions), names of sport clubs, airports, administrative institutions, cinema awards and a few other minor name categories.

Among these Proper Names 30 % of red links are people’s names. The biggest group of these names belong to tennis players.

Interestingly, a great part of red links in Ukrainian Wikipedia (at least as represented by our sample) are not in Ukrainian and many are spelled in other than Cyrillic script. The represented languages are English (e.g. ’John Wiley & Sons’), Russian (e.g. ’Демографический энциклопедический словарь’ (’Demograficheskij entsiklopedicheskij slovar’)), Latin (e.g. ’Idaea serpentata’) and Japanese. Moreover, 31 % of the sample is spelled in Latin script. Among them are red link titles in English, Latin and Ukrainian spelled in Latin script.

The data also has some innate characteristics which were obstacles for retrieval and pre-proccessing steps and which we had to take into account while building our model.

The first is double redirections — redirection pages which redirect to next redirections. For example a page ’Католицизм’ (’Katolytsyzm’) redirected to ’Католицтво’ (’Katolytstvo’) that in turn redirected to ’Католицька церква’ (’Katolyts’ka tserkva’) which is the only existent article here. Fortunately these double redirections are constantly checked and cleaned by Wikipedia users or bots. By the time of writing these lines the redirections mentioned above were already removed and all of them redirected directly to the full article ’Католицька церква’ (’Katolyts’ka tserkva’).

The second type of noise in data is typos in the red link titles. For example ’Панчакутек Юпанкі’ (’Panchakutek Jupanki’) is really ’Пачакутек Юпанкі’ (’Pachakutek Jupanki’), ’Сувалцьке воєводство’ (’Suvalts’ke voyevodstvo’) must be ’Сувальське воєводство’ (Suval’s’ke voyevodstvo). It also goes for other mistakes in writing red links (e.g. ’Неґрська раса’ (’Negrs’ka rasa’) instead of ’Негроїдна раса’ (’Nehroyidna rasa’)). The dangerous thing here is that articles for these red links really exist in the Ukrainian Wikipedia but are not recognized because of the typos. Such and other ’false’ red link titles were revealed during the creation of the ground truth and removed from the dataset regarded as noise. As a result, we obtained a clean sample of 3 171 red links.

4.1.4 Candidate pairs generation. This step is based on the work of our team for Mining Massive Datasets course project at Ukrainian Catholic University on Summer 2018 (Final project for the Mining Massive Datasets course at the Ukrainian Catholic University, 2018). In the project a candidate set was retrieved for English red links among Ukrainian pages. In our work we do it vice versa in terms of Wikipedia editions. For each Ukrainian red link of our sample we have retrieved a set of articles from English Wikipedia which is more probable to contain an entity a red link refers to. Thus, it is called a candidate set and this step is called a Candidate Entity Generation. Our approach to candidate generation is based on common links comparison. The measure of similarity chosen is Jaccard score.

As an input data for retrieving future candidates we use English Wikipedia articles which do not have correspondent pages in Ukrainian Wikipedia yet. We calculate Jaccard score similarity between red links and each of non-translated to Ukrainian English articles according to this formula

$$S_{EU} = \frac{E \cap U}{E \cup U} \tag{6}$$

where E is a set of incoming links for English non-translated articles[22] and U is a set of incoming links for Ukrainian red links. With this approach we obtain the similar articles to our red links ranked from the most similar according to the Jaccard similarity

[18] https://github.com/olekscode/Power2TheWiki

[19] Wikipedia dump from the 20th of September 2018

[20] the dump of the same time

[21] 32 GB of RAM; i7-3930K CPU @ 3.20GHz; GeForce GTX TITAN; 6 GB of VRAM; 256 GB of SSD; Ubuntu 18.04

[22] only those incoming links which have correspondent pages in Ukrainian Wikipedia

red link	candidate
Емад Мотеаб	Mengistu Worku
Емад Мотеаб	Luciano Vassalo
Емад Мотеаб	Wael Gomaa
Емад Мотеаб	Mudashiru Lawal
Емад Мотеаб	Ali Bin Nasser
Емад Мотеаб	Emad Moteab
Емад Мотеаб	James Pritchett (footballer)
Емад Мотеаб	Federation of Uganda Football Associations

Table 1: Generated candidate pairs. Part

score. All the links which have 0 similarity score with a red link are dropped, and we choose pairs which have more than 20 incoming links in common to create a sample of the most popular red links and their candidates in English Wikipedia. In this way a set of pairs 'red link – candidate' is built.

A part of these tables is given in table 1. A size of this set is 2 957 927 red link-candidate pairs for 3 171 red links.

This dataset is unbalanced as for each red link there are about 1 000 candidates among which either only one true candidate or no true candidate is present. It leads to particular ways of managing it in further research.

4.1.5 Creating ground truth. Ground truth for the red link data sample was not provided and no automatic tools for getting it was available. Therefore, we manually created this ground truth in the following way: Ukrainian red link title was searched through Wikipedia and Google search engines. If no appropriate results were found we translated the title in Russian, English or French and repeated the search. The possible results were:

1. English Wikipedia article that was searched.
2. Corresponding Wikipedia pages in other languages.
3. Wikidata item which contains a list of Wikipedia articles in different editions.
4. Wikispecies[23] article from a Wikimedia project aimed to develop a catalogue of all species which also gives a link to English Wikipedia if it exists.

In the process of creating the ground truth for the sample we faced other specific features of the dataset that made the evaluation more difficult. They are the following:

- Different names for a single entity (e.g. 'Білозубкові' ('Bilozubkovi') and 'Білозубки' ('Bilozubky')). It also reveals red links that already have correspondent existent articles in the considered Wikipedia edition.
- Ambiguity. It is hard to find the right correspondence to a red link title just by the name (e.g. 'Austin', 'Guilford', 'Йонас Свенссон' ('Jonas Svensson')). In this context it is often useful to point to a disambiguation page. And evidently more information than just a title is required for matching.
- Red links which by the time of checking for ground truth already became full articles in the considered edition.
- Correspondences that were found by the time of checking for ground truth became deleted articles.

Finally, we labeled 3 171 Ukrainian red links with ground truth.

4.1.6 Train and test sets. A usual random split into a train and a test set based on a fixed partition of the whole dataset is inappropriate in our case. It would divide a candidate set for a single red link between a train and a test set. Therefore, we first randomly split a dataset of unique red links (instead of pairs) in fraction 80 % for a train set and 20% for a test set. And only after that we combine these red links with their candidates. Thus, we obtain a train set of 2 337 270 pairs and a test set of 620 657 pairs.

5 EXPERIMENTS

5.1 Similarity Metrics

Here we present the results of the applied similarity metrics to our data. First, we present our BabelNet baseline and then compare the results of our similarity metrics with the baseline. The metrics are the following: Jaccard similarity measure on incoming and concurrent links of parts of Wikipedia graph, similarity of embedded graphs in Ukrainian and English Wikipedia, Levenshtein edit distance between red links titles and candidate article titles, cosine similarity between embedded titles of Ukrainian red links and English candidate titles. On a train set we search the best parameters for each metrics (threshold and a number of top candidates within which we evaluate the results). Then we validate this results on our test set (620 657 pairs for 635 unique red links). Also, we experiment on how these independent similarity metrics can work together. We apply a linear model on a set which combines the results of our independent models (except for the BabelNet baseline) and validate it on a test set which consists of three candidate pairs for each red link (1 905 pairs for 635 unique red links). The final results on similarity metrics are shown in table 2.

5.1.1 Evaluation metrics. To evaluate and compare the results of different similarity metrics we use F_1 measure. It is calculated on the basis of precision and recall which are defined in terms of true positives (TP), false positives (FP) and false negatives (FN). In our project true positive is an article in English Wikipedia which corresponds to the red link we consider and is marked as such. For example, for the red link 'Анкона (футбольний клуб)' ('Ankona (futbol'nyj klub)') the correspondent English article 'U.S. Ancona 1905' is chosen and it is right. False positive for us is an article in English Wikipedia which does not correspond to the red link we consider but is marked as such. For example, for the red link 'Анкона (футбольний клуб)' ('Ankona (futbol'nyj klub)') the correspondent English article 'Barbara Schett' is chosen and it is false. False negative is an article in English Wikipedia which corresponds to the red link we consider but is marked as not be a such. For example, for the red link 'Анкона (футбольний клуб)' ('Ankona (futbol'nyj klub)') the correspondent English article 'U.S. Ancona 1905' is not chosen but it is right.

5.1.2 BabelNet baseline. We considered BabelNet as a good baseline for our task as it is constructed to a great extent on Wikipedia articles for different languages. Thus, we hypothesized that it might search well for articles through different Wikipedia editions. We

[23] https://species.wikimedia.org/

Similarity metrics	fp	tn	tp	fn	P	R	F_1
BabelNet	4	65	108	456	0.964	0.191	0.32
Incoming links (top1, th=0.26)	385	149	64	37	0.143	0.634	0.233
Concurrent links (top1, th=0.1)	263	360	2	10	0.008	0.167	0.014
Levenshtein (top1, th=0.39)	295	24	41	275	0.122	0.13	0.126
Multi-factored model (top1)	31	1445	339	90	0.92	0.79	**0.85**

Table 2: Evaluation results on similarity metrics as independent models and on a multi-factored model based on the results of the independent ones.
The best F_1 score is in bold

first applied BabelNet multilingual encyclopedia for searching correspondent articles for red links in other Wikipedia editions. In particular, we queried English Wikipedia part of the BabelNet knowledge base. We used both BabelNet Java API and online dictionary to ensure the correctness of results.

The final results of BabelNet approach are presented in table 2. We analyzed what caused so-called false negatives that influence the results for worse. They are the following:

- red link title does not exist in English BabelNet;
- version of BabelNet is older than a wanted page;
- BabelNet doesn't manage with typos;
- red link titles are often written in other than English languages. So first translation to English is needed.

There is a big field for improvement for this task especially for Ukrainian red links problem. Thus, we suggest it as our baseline.

5.2 Graph-based experiments

5.2.1 Calculating incoming links in common. Earlier Jaccard similarity measure served us to select the sample of the most popular red links and create a candidate set. Now it is used as a feature to find the most probable correspondent page for a red link among the candidates. The results obtained on a test set with the best parameters are shown in table 2.

Another graph characteristics we use is concurrent links for a red link and for candidates. Concurrent links are all the links that occur in the same page as a target link. We evaluate Jaccard similarity metrics on concurrent links as an independent model as well. With derived threshold of 0.1 and top 1 number of candidates we obtain the results for a test set shown in table 2.

Graph embedding techniques did not work out for our data due to our current technical resource limit pointed out earlier.

5.2.2 Word-based similarity model results. Word-based feature that we can use to compare red links and candidate articles is their titles.

First, to apply Levenshtein edit distance on pairs each red link was transliterated into Latin script by means of the bi-directional transliterator for Python[24]. Then it was matched with each of all its candidates. The candidate page which has the lowest edit distance with the red link is considered to be its correspondence page in English Wikipedia. Applying Levenshtein edit distance as an independent model we obtain the results shown in table 2.

Our next approach was cross-lingual embeddings. We applied it to a sample of 100 red links. As a result, no correspondent items were found correctly but all suggested items are of the same topical meaning as respective red links.

5.3 Multi-factor Similarity-based Model

Finally, we combined all our similarity metrics as features under the logistic regression model. We appealed to this machine learning algorithm because of two reasons. The first is to make our results on similarity metrics easy to interpret. The second is to improve the results of independent similarity metrics to solve our task.

Since our train and test sets are highly unbalanced, we reduced a number of less probable candidates. We assume that each similarity metrics choose the most probable candidate among others. Therefore, we compose our refined train and test sets from the most probable candidates chosen by each similarity metrics that worked out on our data: namely by Jaccard score on incoming links, Jaccard score on concurrent links and Levenshtein edit distance on titles. This way from the train set of 2 337 270 pairs we obtain a train set of 7 608 pairs, from the test set of 620 657 pairs we obtain a set one of 1 905 pairs. In other words, now each Ukrainian red link has three candidates from English Wikipedia which are the three most probable candidates according to the applied similarity metrics before.

As features to train our model we use the scores of three mentioned similarity metrics and also exploit the results of BabelNet. We use scikit-learn library[25] for training and evaluation of a logistic regression model. The result of applying logistic regression with the chosen features on our data is $F1$ score 85 % (with precision = 92 % and recall = 79 %).

We also estimate importance of each feature for the model. This estimation is calculated based on absolute values of the coefficients derived by the logistic regression model. The results on feature importance are presented in figure 2.

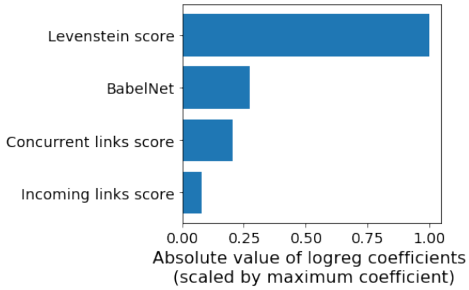

Figure 2: Relative feature importance for logistic regression model

In the process of our project we experimented with subsamples of different entity groups from our data: proper names, names of people, titles in Latin script. For all of them Levenshtein edit distance works as the best independent similarity model but the second best metrics for these groups is different. BabelNet is good

[24] https://pypi.org/project/transliterate/

[25] https://scikit-learn.org/

only for Latin script titles. In other cases the second best result belongs to incoming links similarity model.

As we see, word based similarities are the most important in our approach. We hypothesize that the reason for that is that most of the red links correspond to nouns, that does not change significantly across languages. This feature is especially boosted given that we are transliterating from Cyrillic to Latin script in order to run this comparison. Future work will be necessary to understand how this metric is affected by pairs of scripts that are more difficult to translate (for example from some Asian scripts to English).

6 CONCLUSIONS

In this work we presented a solution for resolving red links in Ukrainian Wikipedia. It is the first and for now the unique prototype for a tool of matching Ukrainian red links with existed articles in English Wikipedia. All the code is released on github and is open for further use.

We presented a step-by-step Named Entity Linking pipeline, from retrieving data and creating datasets to applying a machine learning model to resolve red links. The created datasets have been published. The first is a dataset of the most frequent 3 171 Ukrainian red links. They occur in 20 or more articles which have corresponding articles in English Wikipedia. The second dataset consists of 2 957 927 pairs of the retrieved red links and their candidate pages from English Wikipedia. The dataset is open for further use and research. We supplied it with statistics and its characteristics analysis which is meant to boost further experiments.

Exploring related projects in the Wikimedia community we found out that no significant previous work had been done to solve this problem. We introduced BabelNet knowledge base as a tool for translating red link items into other languages. As a result we presented its powers for resolving Ukrainian red links through English BabelNet knowledge base.

We stated the problem of resolving red links as a Named Entity Linking task. After exploring background work in this field we defined the methodology which is suitable to our task and the main components necessary for the experiments.

We assumed that matching red links with items in other Wikipedia editions could be solved through Wikipedia graph properties and word properties of their titles. Based on that assumption we chose several similarity metrics to find the correspondent page as most similar to a red link by the mentioned properties: Jaccard similarity on incoming links to red links, Jaccard similarity on links which occur in the same pages as red links, Levenshtein edit distance on titles of red links and their candidate pages. Interestingly, we found that simple metrics such as a Levenshtein distance performs better than other more sophisticated approaches. We hypothesize that this is due the morphological characteristics of red links.

In this work we also described the failed experiments such as graph embedding and cross-lingual embedding. Our experiments showed that the applied graph embedding is not suitable for our task due to our current technical resource limit and the characteristics we build our graph on. With Jaccard measure we calculate the similarity on the same properties with much less resources. As for cross-lingual embedding that is good for matching topics instead of particular correspondent pages, this model could be helpful as a part of an ensemble algorithm. We suggest it as our future work. Finally, we presented results on our multi-factor similarity-based model which combined all previous results of our project. We used logistic regression for a linear model and achieved F_1 score 85 % which is quite good to make a prototype tool for solving this problem.

REFERENCES

[1] Nuha Alghamdi and Fatmah Assiri. 2019. A Comparison of fastText Implementations Using Arabic Text Classification. In *Proceedings of SAI Intelligent Systems Conference*. Springer, 306–311.

[2] Christopher M Bishop. 2006. *Pattern recognition and machine learning*. Springer.

[3] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, 1247–1250.

[4] MS Fabian, K Gjergji, WEIKUM Gerhard, et al. 2007. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *16th International World Wide Web Conference, WWW*. 697–706.

[5] Palash Goyal and Emilio Ferrara. 2018. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems* 151 (2018), 78–94.

[6] Dan Jurafsky and James H. Martin. 2019. Speech and Language Processing. In *Speech and Language Processing*. Third Edition draft.

[7] Sven Kosub. 2019. A note on the triangle inequality for the jaccard distance. *Pattern Recognition Letters* 120 (2019), 36–38.

[8] John M Lee. 2013. Smooth manifolds. In *Introduction to Smooth Manifolds*. Springer.

[9] Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 288–297.

[10] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2017. A survey of cross-lingual word embedding models. *arXiv preprint arXiv:1706.04902* (2017).

[11] Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27, 2 (2014), 443–460.

[12] Ehsan Sherkat and Evangelos E Milios. 2017. Vector embedding of wikipedia concepts and entities. In *International conference on applications of natural language to information systems*. Springer, 418–428.

[13] Li Yujian and Liu Bo. 2007. A normalized Levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence* 29, 6 (2007), 1091–1095.

[14] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 483–491.

[15] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 425–434.