SWLab @ Covid-19 MLIA Information Extraction Task

Sandro Gabriele Tiddia[1], Diego Marcia[2], Manuela Sanguinetti[2], and Maurizio Atzori[2]

DMI - Department of Math/CS,
University of Cagliari, Italy
[1]sandrogabrieletiddia@outlook.it
[2]{first.last}@unica.it

Abstract. In this report we describe the system used to participate in the Information Extraction task (Task 1) at the Covid-19 MLIA evaluation campaign. The task consists of a three-round evaluation process aimed at identifying Covid-related information in raw texts. Our proposal is based on automatically expanding a user-provided small seed of words representing a class (e.g., "coronavirus", "flu" and "pneumonia" for *symptoms and diseases*). Our prototype system obtained low results in this task, but alternative solutions have been spotted to overcome the current limitations.

1 Introduction

The Covid-19 MLIA @ Eval campaign [6] invites participants to develop systems capable of processing information related to Covid-19 for an adequate identification of the phenomenon and possible actions aimed at its contrast. For this purpose, the Information Extraction task (Task 1) of this campaign is open to submissions in which, given an input text, text spans are identified and assigned to one of the six macro-categories defined below:

- drug names, treatments, general intervention (tagged as *drug-trt*)
- signs, symptoms, diseases (*sosy-dis*)
- findings, efficacy of treatments (*findings*)
- tests (*tests*)
- behaviors, everyday life actions (*behavior*)
- legal dispositions, regulations (*legal-reg*)

For all the details related to the task and its organization, we refer to the overview report provided by the organizers [7]. In this report[1], we describe the

Covid-19 MLIA @ Eval Initiative, http://eval.covid19-mlia.eu/.

[1] Please note this is an ongoing work currently describing only Round 1 (out of 3 rounds in total) of Task 1 at the Covid-19 MLIA evaluation campaign.

system used by the SWLab team to participate in this task. More specifically, we adopted an alternative approach with respect to the classic information extraction systems, rather considering the problem as a *set expansion* task. Set expansion is a task that consists in expanding a small set of entities belonging to the same semantic class, defined as seeds, with new entities belonging to that class. More formally, given a seed set $S = s_1, ..., s_k$ of a class $C \supset S$, and an unlabeled text corpus T, set expansion consists in finding all members of class C [8]. Within the specific context of this task, the semantic classes are the six macro-categories proposed by the organizers and also mentioned above. Each seed set thus consists of hand-crafted small tuples of entities from one of these categories, while the candidate set members are entities extracted from the raw texts provided as training and test data by the task organizers.

While being aware that such an approach cannot obtain state-of-the-art results, the aim of our participation in this task is to explore the effectiveness of set expansion techniques in sequence labeling tasks, especially in the absence of pre-annotated training data.

Next sections describe the system used for this purpose, along with the results obtained in the first round of the evaluation, and a discussion on the main system's shortcomings and possible future directions.

2 System Description

For the set expansion task, we resorted to the OKgraph library [2], developed at the University of Cagliari and publicly available in a GitHub repository[2]. The library has been conceived as a tool to perform various unsupervised NLP tasks, besides set expansion [5], such as set labeling, relation extraction and labeling, and hypernym discovery [3].

For the set expansion task, the OKgraph library features a number of unsupervised algorithms exploiting word embeddings with different strategies [4]: nearest-neighbor search over centroid, depth search, 2WC, top-k means. For the MLIA evaluation campaign, we focused on a simple approach: nearest-neighbor search over centroid with incremental boost. Given a small seed of words or small phrases belonging to a given category (e.g., "covid-19", "flu" and "high fever" for the category *signs, symptoms, diseases*), the algorithm computes the word embeddings (we tested word2vec embeddings) over a given corpus of unannotated text. The centroid of the word embeddings belonging to the seed is computed, and used to find the top-k nearest neighbours, that are expected to represent co-hyponyms of the given seed set, therefore belonging to the same class. One difference w.r.t. classic classification approaches is that humans are only expected to find a small seed (order of 3-5 examples), which makes it suitable when no large tagged datasets are available, as required by standard supervised machine learning approaches.

For a better performance of the set expansion task we defined 8 classes as a more fine-grained subdivision of the 6 macro-categories. Based on preliminary

[2] `https://github.com/atzori/okgraph`

results, we eventually identified a small seed for 4 of the 8 classes in order to perform the set expansion task and populate the classes with similar entities, covering 3 of the 6 macro-categories. The limit to the number of results obtainable from the task has been set to 200 entities. This relatively low threshold we used for the submitted runs was chosen with the aim of avoiding a higher occurrence of non-related entities, that is, to reduce false positives. The seeds used in the set expansion task, along with their classes, are the followings:

- **drug** from *drug-trt*
  - en-seed: {improvac, pemetrexed lilly, protopic}
  - it-seed: {improvac, pemetrexed lilly, protopic}
- **sosy** from *sosy-dis*
  - en-seed: {breathing difficulty, disorientation, blindness}
  - it-seed: {respirazione difficoltosa, disorientamento, cecità}
- **dis** from *sosy-dis*
  - en-seed: {tardive dyskinesia, diabetes mellitus, cardiomyopathy}
  - it-seed: {discinesia tardiva, diabete mellito, cardiomiopatia}
- **test** from *tests*
  - en-seed: {screening, ct scan, mammography}
  - it-seed: {screening, tomografia computerizzata, mammografia}

The entities from the populated classes are searched inside the training and test dataset (both unlabeled) to find all their occurrences. For every found occurrence, an annotation is generated using the category associated to the entity class. To make the search more efficient, every line from the files in the dataset is indexed in a preliminary phase, storing the line content along with its offset from the beginning of the file and the path of its text file. The index is used to easily retrieve the lines in which the entities occur, calculate the starting and ending offset of the match and write the annotation in the correct annotation file. Indexing was performed via the Whoosh library[3].

Prior to feeding the data to the OKgraph library, we created a *unified corpus*, by concatenating all the text files elected for word embedding extraction (see section 2.1 for the three different options). Then, the files in the test set were indexed and the whole unified corpus was passed to OKgraph.

2.1 Round 1

We submitted our results for two out of the seven languages available for the task, i.e., English and Italian. As described above, the set expansion module relies on pre-trained word embeddings to identify the most similar entities in the given input text. We thus tested the expansion algorithm with three different options to create the embeddings:

1. using the texts from the provided training set only
2. using the training data with the addition of 1GB plain text Wikipedia dump

[3] `https://github.com/mchaput/whoosh`

3. using the texts from the options above with the addition of the test data

The plain texts from the Wikipedia dumps were extracted using WikiExtractor [1], a Python standalone script that extracts and cleans text from Wikipedia snapshot data. Given a (possibly compressed) XML dump from Wikipedia[4], WikiExtractor cleans each Wikipedia page's text from HTML and MediaWiki markup tags. The script creates a number of folders, each containing several files with the plain text from multiple articles. The whole text for each article is still encapsulated inside a pair of opening/closing `<doc>` markup tags. Files and folders are all of similar size (∼1 MB per file, 100 files per folder).

For the English test set we submitted three runs – the maximum allowed for each language – each one corresponding to the settings outlined above. For the Italian test set, since the first setting did not produce any result, we just submitted the results produced using the two embedding models obtained with the second and third setting.

3 Results and Discussion

In this section we report the results obtained for the submitted runs.

3.1 Round 1

Since no Italian gold standard has been provided due to the lack of submissions from other teams, only the English sets were evaluated in this round.

As described in the task overview, the gold standard annotations for English have been manually produced by the organizers, who annotated a selection of 9 files from the test dataset, choosing the most frequently annotated files by the four participants. In total, 1740 words or sentences have been tagged, generating a dictionary of 269 unique words/sentences (case sensitive).
As reported by the organizers, 67.1% of the annotations come from the dictionary subset *COVID-19, Covid-19, covid19, COVID19, Coronavirus Disease 2019, coronavirus, Coronavirus.*

The organizers also generated a ROVER dataset of 52 files, obtained by merging annotations which have been produced by at least 2 of the 4 participant teams. A total of 15133 annotations make up this dataset, with a dictionary of 1963 unique words/sentences (case sensitive).

We were able to produce 11 unique annotations (9 unique annotations ignoring case): for the first run, a total of 8 annotations have been found, while for the second and the third run our code found 9 annotations; in each run, our code produced 7 unique annotations (case sensitive).

Rover. These are the six annotations from our submissions which were included in the Rover dataset (i.e., they have also been found by at least another team):

[4] `https://dumps.wikimedia.org/backup-index.html`

Run 1	3657-ab.ann	sosy-dis	147041 147052	Mood swings
Run 2	3706-aa.ann *	tests	2967 2987	clinical examination
Run 2	3708-aa.ann	sosy-dis	5626 5648	Breathing Difficulties
Run 3	3657-ab.ann	sosy-dis	147041 147052	Mood swings
Run 3	3706-aa.ann *	tests	2967 2987	clinical examination

* One occurrence in ROVER, two in our submission

Gold Standard. This the only annotation in common with the gold standard:

Run 2	3708-aa.ann	sosy-dis	5626 5648	Breathing Difficulties

Error Analysis. As described in the previous section, our system produced unsatisfactory results, with a precision $P = 0.00$ for all the annotation categories. The primary reason for these results was that the system produced very few annotations for the test set. Although low performance was partly expected considering that unsupervised set expansion was used for a different task, we further analyzed the system's output in order to identify the main shortcomings of our approach. Below we provide an overview of the main causes we spotted with our analysis.

Index Lookup Unexpected Behaviour. Upon investigating the reason for such a low annotation count, we found out we had overlooked a setting in one of the library functions, used for looking up matches in the corpus index. This function exposes a default behavior that limits the number of retrieved matches. In the following table, we show our results with the correct setting, as counted by the evaluation tool distributed by the organizers[5]. As it can be seen, changing this setting significantly increased the number of predictions, but not their quality overall.

Run	Predictions	behavior (n=228)	drugs (132)	find. (1)	legal (160)	sosy (1173)	tests (46)	overall
Run #1	49	.000	.000	.000	.000	.2632	.0333	.1224
Run #2	71	.000	.000	.000	.000	.2400	.0217	.0986
Run #3	63	.000	.000	.000	.000	.2174	.0250	.0952

Table 1. Number of predictions and results (in terms of Precision) for each category (behavior, drugs/treatments, findings, legal rules, sign or symptoms/diseases, tests) and globally (overall) for each run, using the gold standard annotations (9 files). The number of annotations per category in the reference is presented between parentheses.

[5] `https://bitbucket.org/covid19-mlia/organizers-task1/src/master/ground-truth/round1/`

Seed sets. Another possible reason for the low performance in this task lies in the nature of the seeds created for the three runs. As a matter of fact, the entities included in the seed sets, though pertaining to the six categories of the task, do not strictly relate to the topic the task focused on, namely Covid-19. As also pointed out in the task overview, the identification of entities more directly related to this topic was one of the ultimate goals of the task. We therefore did some post-evaluation experiments to test the system's results using seeds from more to less relevant to Covid-19. This is in turn to verify both the possible results obtained with a more Covid-related seed set and how system's results might be affected by the different seeds given as input.

For illustrative purposes, we performed our experiments for the *sosy-dis* category only. More specifically, we first ran the system using Covid-related entities as seeds for the class *diseases*, i.e. *sars, coronavirus, pneumonia*. This allowed us to verify the system's results using entities semantically closer to the main topic of the task. We then iteratively ran the same system with different seeds at each iteration, in order to assess the second main question, i.e. the system's sensitivity to different input seeds. To generate the seeds, we randomly selected from the expanded set of the first attempt 20 new tuples including the entities reported in Table 2.

The system settings were the same used for Run #3 also described in Section 2.1. Even in this case, we evaluated all the results against the gold standard, using the tool provided by the organizers. Table 2 shows the scores obtained with each seed.

The experiment showed two divergent results: on the one hand, as expected, using more semantically-related seeds significantly improved the output quality, at least for the tested category. On the other hand, the randomly-created seeds produced much lower results, in terms of F-score, with respect to the first experiment, but they are very similar among them, reporting a very low variance of the F-score ($\sigma^2 = 0.0004$). Such uniformity of results is mostly due to the very low Recall obtained with each seed (as opposed to Precision, ranging from 0.28 to 0.88), which in turn negatively affects the overall F-score. These results seem once again to suggest the crucial role played by the accurate selection of seeds with respect to the entities to be identified. The automatically generated seeds contain types of diseases that are only partially or not at all relevant to Covid-19, and this might explain the very high frequency of false negatives in the results produced, which is much lower in the first experiment, as also highlighted in Table 2.

Tuning k-neighbours parameter. As previously stated in the system description, our approach aims at performing set expansion by finding the top-k nearest neighbours to the seed's centroid. For our submission, the k parameter has been arbitrarily set to 200. With this final batch of experiments, we then aimed to verify whether more optimal values of k could be found, using the third run setting and the Covid-related seed set introduced in the previous experiments,

[5] This variance increases to $\sigma^2 = 0.0089$ if the Covid-related seeds are included.

Seeds	Precision	Recall	F-score
sars, coronavirus, pneumonia	0.5475	**0.4092**	**0.4683**
poliomyelitis, peritonitis, basal_cell_carcinoma	0.5625	0.0115	0.0226
typhoid, staphylococcus_aureus, diseases_like	0.5882	0.0222	0.0429
merscov, viral_infections, japanese_encephalitis	0.6531	0.0295	0.0565
encephalitis_lethargica, gastroenteritis, communicable_diseases_such_as	0.6957	0.0208	0.0405
amyotrophic_lateral_sclerosis, tuberculosis, encephalitis	0.6667	0.0176	0.0344
herpes, chikungunya, h1n1_influenza	0.6471	0.0143	0.0280
typhoid_fever, malaria, sexually_transmitted_infections	0.6757	0.0444	0.0833
smallpox_measles, pneumonitis, malignancies	0.2857	0.0024	0.0047
pneumonia, chickenpox, salmonellosis	0.6667	0.0236	0.0457
polio, cardiovascular_diseases, tetanus	0.7442	0.0294	0.0566
yellow_fever, diphtheria, sexually_transmitted_infections	0.6842	0.0478	0.0894
merscov, respiratory_infections, diseases_such_as	0.6222	0.0258	0.0495
herpes_simplex, cytomegalovirus, rubella	0.8846	0.0299	0.0578
pseudomonas_aeruginosa, causative_agent, human_immunodeficiency_virus_hiv	0.5000	0.0230	0.0440
amyotrophic_lateral_sclerosis, liver_damage, disease	0.2966	0.0309	0.0560
campylobacter, avian_influenza, measles_virus	0.5455	0.0221	0.0424
measles_virus, skin_cancer, f_necrophorum	0.7500	0.0195	0.0381
amyotrophic_lateral_sclerosis, viral_infection, flu	0.7059	0.0276	0.0532
infections_such_as, leprosy, cushing_s_syndrome	0.4800	0.0135	0.0263
pulmonary_edema, birth_defects, gastroenteritis	0.4286	0.0035	0.0070

Table 2. Annotation results for the *sosy-dis* category with different seed sets for the class *diseases*. The first row shows the results obtained using Covid-19 related seed entities, while the other ones below reports the results obtained with 20 randomly-generated seed sets.

i.e. *sars, coronavirus, pneumonia*. Table 3 presents the evolution of the score metrics for the class *diseases* only, with incremental values for the k parameter.

The optimal results in these experiments are found with $k = 175$, which provided the higher F-score with respect to the other values. Also, by setting the parameter to $k = 20$, we get high-Precision results, but very low Recall and F-score, while increasing the parameter value – therefore the overall number of annotations – Recall increases at the expense of Precision, though very slowly.

As a general remark, what we can observe from this one, as well as the previous experiments, is that despite the major changes made to the system settings, the set expansion techniques we adopted in this context are more likely to produce high-Precision, but low-Recall results, thus reducing the possibility to provide broad-coverage, and state-of-the-art results for information extraction.

More systematic studies can be carried out in the near future to further support the qualitative analyses presented here.

k	Annotations	Precision	Recall	F-score
20	306	**.8105**	.2166	.3418
100	656	.4893	.2723	.3499
125	658	.4894	.2731	.3506
150	665	.4902	.2763	.3534
175	668	.4895	.2769	**.3537**
200	675	.4844	.2769	.3524
300	685	.4774	.2769	.3505
500	729	.4540	**.2798**	.3462

Table 3. Evolution of the scores wrt incremental k values.

4 Conclusions and Future Work

In this report we described the system used to participate in Task 1 of the Covid-19 MLIA initiative. The system tackles a typical Information Extraction task resorting to set expansion techniques, i.e. starting from small sets of entities, each one related to the six categories defined for this task, identify in the text other entities or text spans related to these categories and label them accordingly. It is worth pointing out that the training set provided by the organizers consisted of several raw-text files devoid of annotation layers of any kind, which is what mainly motivated the use of unsupervised techniques for the purposes of the task. In the first-round evaluation phase our system did not produce the desired results. We thus carried out a number of small-scale experiments to verify the main weaknesses of the whole approach. The experiments showed that a proper selection of the seed entities and of the number of nearest neighbours to the seed's centroid clearly have an impact on the system's results, but more in-depth studies are required to verify, first and foremost, whether this also applies to the other categories not taken into account in our post-evaluation experiments, but also how and to what degree these two factors can be automatically defined or rather a human-in-the-loop process is a much desired option.

Furthermore, the extrinsic evaluation of our set expansion system through the output and the metrics of an Information Extraction task highlighted the main tendency of the former to produce higher-Precision and lower-Recall results, when applied for the latter task. Once again, further studies are needed to verify whether this is a consistent behavior and, if so, how to improve the results in terms of Recall as well.

Acknowledgements

Supported in part by MIUR PRIN 2017(2019-2022) project *HOPE - High quality Open data Publishing and Enrichment* and Regione Sardegna project *ADAM: Activity recognition in Dual Acquisition Mode: analysis, feature extraction and classification of actions and activities.*

References

[1] Attardi, G.: Wikiextractor. https://github.com/attardi/wikiextractor (2015)

[2] Atzori, M.: The need of structured data: Introducing the okgraph project. In: Crestani, F., Noia, T.D., Perego, R. (eds.) Proceedings of the 8th Italian Information Retrieval Workshop, Lugano, Switzerland, June 05-07, 2017, CEUR Workshop Proceedings, vol. 1911, pp. 121–124, CEUR-WS.org (2017), URL http://ceur-ws.org/Vol-1911/22.pdf

[3] Atzori, M., Balloccu, S.: Fully-unsupervised embeddings-based hypernym discovery. Inf. **11**(5), 268 (2020), https://doi.org/10.3390/info11050268, URL https://doi.org/10.3390/info11050268

[4] Atzori, M., Balloccu, S., Bellanti, A.: Unsupervised singleton expansion from free text. In: 12th IEEE International Conference on Semantic Computing, ICSC 2018, Laguna Hills, CA, USA, January 31 - February 2, 2018, pp. 180–185, IEEE Computer Society (2018), https://doi.org/10.1109/ICSC.2018.00033, URL https://doi.org/10.1109/ICSC.2018.00033

[5] Atzori, M., Balloccu, S., Bellanti, A., Mameli, E., Usai, S.R.: Okgraph: Unsupervised structured data extraction from plain text. In: Agosti, M., Buccio, E.D., Melucci, M., Mizzaro, S., Pasi, G., Silvestri, F. (eds.) Proceedings of the 10th Italian Information Retrieval Workshop, Padova, Italy, September 16-18, 2019, CEUR Workshop Proceedings, vol. 2441, pp. 30–31, CEUR-WS.org (2019), URL http://ceur-ws.org/Vol-2441/paper19.pdf

[6] Casacuberta, F., Ceausu, A., Choukri, K., Declerck, T., Deligiannis, M., Di Nunzio, G.M., Domingo, M., Eskevich, M., Ferro, N., García-Martínez, M., Grouin, C., Herranz, M., Jacquet, G., Papavassiliou, V., Piperidis, S., Prokopidis, P., Zweigenbaum, P.: The Covid-19 MLIA @ Eval Initiative: Developing Multilingual Information Access Systems and Resources for Covid-19 (2020)

[7] Grouin, C., Declerck, T., Zweigenbaum, P.: The Covid-19 MLIA @ Eval Initiative: Overview of the Information Extraction Task (2020)

[8] Pantel, P., Crestan, E., Borkovsky, A., Popescu, A.M., Vyas, V.: Web-scale distributional similarity and entity set expansion. In: EMNLP 2009 - Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: A Meeting of SIGDAT, a Special Interest Group of ACL, Held in Conjunction with ACL-IJCNLP 2009, pp. 938–947 (2009)