Extracting statistical mentions from textual claims to provide trusted content

Tien Duc Cao, Ioana Manolescu, Xavier Tannier

Extracting statistical mentions from textual claims to provide trusted content

Tien Duc Cao[1,2], Ioana Manolescu[1,2], and Xavier Tannier[3]

[1] Inria Saclay Île-de-France
[2] LIX (UMR 7161, CNRS and École Polytechnique)
[3] Sorbonne Université, Inserm, LIMICS (UMRS 1142)

Abstract. Claims on statistic (numerical) data, e.g., immigrant populations, are often fact-checked. We present a novel approach to extract from text documents, e.g., online media articles, mentions of statistic entities from a reference source. A claim states that an entity has certain value, at a certain time. This completes a fact-checking pipeline from text, to the reference data closest to the claim. We evaluated our method on the INSEE dataset and show that it is efficient and effective.

1 Introduction

With the increase of disinformation in online media, social networks and the Web in general, we witness a strong interest in computational fact-checking, defined as a set of computer-assisted techniques capable of assessing the truthfulness of a given statement [4]. In this context, computational fact-checking is a many-stage pipeline, whereas (*i*) claims are extracted from text, (*ii*) possible sources of reference are identified, (*iii*) a check is made combining automated and manual means; (*iv*) an interpretation is produced.

In this paper, we focus on steps (*i*) and (*ii*). We use data from French national institute for statistics and economic studies (INSEE) as an example as high-quality, trustful reference database. In previous work, we have **extracted tens of thousands of RDF graphs** out of INSEE statistic tables [2][4]. We also developed a novel **keyword search algorithm** which, given a set of search terms, e.g. *"unemployment"*, *"Île-de-France"*, *"2018"* locates the RDF nodes corresponding to the most relevant table cells [3][5].

In this work, we describe the last missing step of our system: the **extraction of claims referring to statistical mentions from text sources**. This step allows to automatically formulate the search queries which our system [3] can solve against the RDF corpus we gathered [2]. Our whole system can help fact-checking journalists to find checkable claims in massive text sources, as well as the closest reference datasource value for the given claim. Based on these, the journalists can choose the truth label which seems most appropriate.

The architecture of the system is presented by Figure 1. From the publication context of statistic data (the text in header of statistics tables) we extract a set

[4] https://gitlab.inria.fr/tcao/insee-search/blob/master/insee-rdf.ttl.gz
[5] The search algorithm is deployed online at http://statsearch.inria.fr

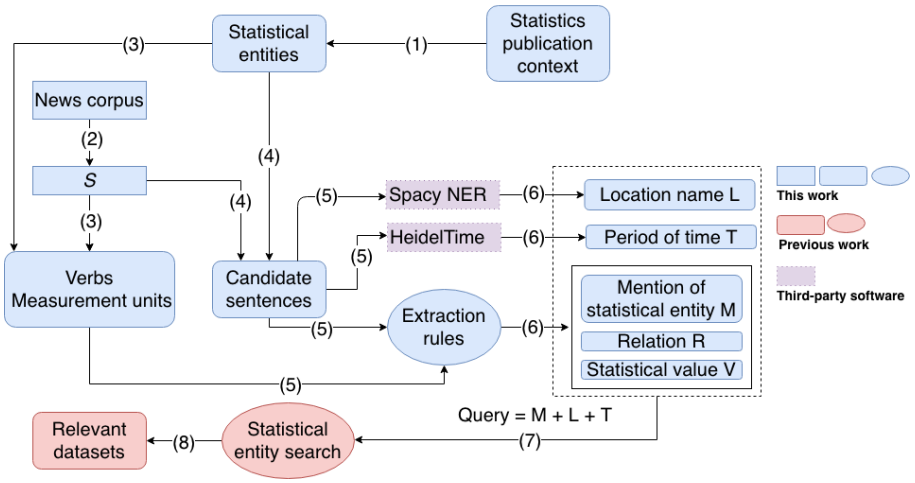

Fig. 1. Main processing steps of our statistical claim extraction method.

of **statistical entities** (step (1) in the figure), those whose reference values are known in the statistic dataset for some time periods and/or geographical area, such as *"unemployment", "youth unemployment", "unemployment in Aquitaine in 2015", "gross domestic product"*. From 111,145 tables published by INSEE, we have obtained a total of 1,397 statistic entities, as we detail in Section 2.1.

We have built a **text corpus** which we selected with an interest in topics that INSEE studies. We focused on news articles from three French newspapers, and because most INSEE metrics refer to the economy domain, we looked only for articles on such topics, by using URL keywords or an LDA [1] topic selection[6]. From these articles, we have extracted (step (2)) 322,873 sentences containing at least one numerical value. From now on, we will refer to these sentences as S. From S, we extract (step (3)) all the verbs which state a numerical value, e.g., "amounts to", "is worth", "decreases" etc., as well as all the measurement units, e.g., "people", "euros", "percentage" etc.

Next, we identify among S sentences the **candidate sentences** which could claim a relationship between a statistical entity and a value. This is done (step (4)) by selecting those S sentences which mention statistic entities. From each candidate sentence, e.g., *"France's public debt fell slightly, by 11.4 billion euros, between the second and third quarters of 2013"*, we extract: **(1)** a mention of statistical entity M, e.g., *public debt*; **(2)** optionally, a location L, e.g., *France*, by extracting geographical places using the spaCy Named Entity Recognition tool[7]; **(3)** optionally, a time period T, e.g., *2013*, extracted using HeidelTime [9]; **(4)** a relation R, e.g., *fell*, connecting M to V in the sentence. R may also be missing, e.g., in a phrase such as *"France's 60 million inhabitants..."*; **(5)** a statistical value V, e.g., *"11.4 billion euros"*.

For each (M, L, T, R, V) tuple extracted as above, the (M, L, T) query is generated (step (7)) and sent to our keyword search algorithm [3]. We omit R

[6] All topics and their keywords are available at https://gitlab.inria.fr/tcao/news-scraper/blob/master/lesechos_topics_all.txt. In this work, we use topics 1, 2, 3, 7.

[7] https://spacy.io/models/fr#fr_core_news_md

Extracted statistical entities	**Frequency**
intensité de la pauvreté (intensity of poverty)	190
nombre d'entreprises (number of companies)	176
taux de pauvreté au seuil de 60% (poverty rate at 60% median wages)	130
chômeurs (unemployed people)	104
excédent brut d'exploitation (Earnings before Interest, Taxes and Amortization)	68
PIB (gross domestic product, GDP)	54
taux de population en sous-emploi (share of people working less than they would like)	54
solde migratoire (net migration)	44
taux de marge (margin rate)	28
taux de pauvreté (poverty rate)	21

Table 1. Sample extracted statistical entities.

in the query since the purpose of extracting R is to confirm the relationship between M and V.

2 Entity, relation and value extraction

We present our approach to extract the components M, R and V.

2.1 Statistical entities

We made a hypothesis of the existence of statistical entities in the headers of statistic tables. For example, one header of table[8] is *"Taux de chômage au T1 2015"* ("Unemployment rate in the first quarter of 2015"). We keep only headers that contain a measurement unit such as euro, %, etc. These headers are usually noun phrases in format *Entity + (Unit)* such as "Unemployment rate in 2015 (in %)". We prefer to rely on table headers and not on table titles and comments, since the latter are longer sentences that could (or could not) contain the entities, and customarily do contain much more irrelevant information. We also filter out possible date time values and their associated prepositions. In the above example, this leads to the snippet "Unemployment rate". A final manual filtering allowed us to weed out some text snippets which do not in fact comprise relevant entities.

We thus obtained 1,397 statistical entities, some of which are presented, with their frequencies from the statistics publication context, in Table 1.

2.2 Relevant verbs and measurement units

We use the annotation S_I to refer to the candidate sentences that contain the word *"insee"*. These sentences are likely to feature a relationship between a mention of statistical entity M as a noun phrase (e.g. "unemployment rate") and a statistical value V as a numerical value, optionally followed by a measurement unit (e.g. "5%").

We used spaCy [7] to collect the syntactic dependency paths connecting M, R and V. For each NOUN node, we located the paths that connect it to a NUM node. Many paths start with (NOUN, `nsubj`, VERB) (a noun is subject of a verb); we refer to them as $Paths_I$. As the relation R of M and V is generally

[8] https://www.insee.fr/fr/statistiques/1288156#tableau-Figure_2

introduced by specific verbs, we collected all the verbs associated with VERB nodes from $Paths_I$. To make sure of the quality of the collected verbs, we filtered manually from the original list to retain 129 relevant ones; in the sequel, we denote them by *I_verbs*. Based on $Paths_I$, we also gathered a set of measurement units by collecting all the NOUN nodes connected to a NUM node via a `nmod` edge (nominal modifiers of nouns or noun phrases). We call this list *I_units*.

2.3 Extraction rules

Given the input sentence i and a statistical entity e, we extract the mention of statistical entity M, the statistical value V and their relation R. If there is no relationship between e and the statistical value, or there is no statistical value in i, we return the value $M = None$. We identify from the dependency tree the statistical entity e and the numerical value(s), as follows.

1. We filter out the year values (e.g. 2018) since we only want to search for the relationship of statistical entity and statistical value.
2. We define the distance $d(n_1, n_2)$ of two nodes n_1 and n_2 in $t(i)$ as the absolute value of n_1's position - n_2's position. For instance, $d(\textit{inflation}, \textit{établie}) = 3$.
3. The distance $D(e, v)$ from e to a numerical value v is the minimum value of $d(e\text{'s first word}, v)$ and $d(e\text{'s last word}, v)$. In case there are more than one numerical values, we select the one that has the smallest $D(e, v)$ as the statistical value of e.
4. We identify the dependency path $p(i)$ that connects **the first word of** e (let's call it s) and e**'s statistical value** (if available), let's call it n. With our sample dependency tree, $p(i)$ = (NOUN, `nsubj:pass`, VERB, `obl`, NOUN, `nummod`, NUM)
5. We look for the node u directly connected to n (the last one before n) in $p(i)$. If u is a noun and there is a `nmod` edge (nominal modifiers of nouns or noun phrases) between u and n, we return $M = None$ in the following cases:
   - u does not appear in *I_units*.
   - u appears in *I_units* and in the input sentence, there is an article or an adposition between s and u.

   On the contrary, we extract the relevant nodes from:

   (a) the first NOUN node s: we identify the nodes that connect to s via `nmod` and `amod` (adjectival modifier) edges, and we collect their subtrees.

   (b) the VERB node *verb*: the subtree of nodes that connect to *verb* via `obl` edge (a nominal dependent of a verb), the leftmost node of subtree must be a preposition among *en, à, dans* and *verb* has to appear in *I_verbs*.

   If the nodes from these subtrees appear in $p(i)$, we do not include them.

All the extracted nodes form the mention of statistical entity M. The statistical value V is composed of n and u. The relation R is composed the nodes from $p(i)$ which do not belong to M and V.

3 Evaluation

Evaluation of the extraction rules We select some statistical entities[9] from the list of statistical entities in Section 2.1. For each entity e we pick randomly 50 sentences that contain e then we split randomly 25 sentences for development set and 25 sentences for test set. Finally there are 200 sentences for each set. If there is no relationship between e and the statistical value, or there is no statistical value in the given sentence, we assign a label *NoStats*. Otherwise we annotate each sentence with e and the relevant phrases (we call these phrases *contexts* of e) to form a mention of statistical entity[10]. For a given sentence, if the extraction rules return $M = None$ and we have the *NoStats* label from the annotated sentence then the extraction is an accurate one. On the contrary, we verify if the extracted M contains e and one of its contexts. In that case, the extraction is also accurate. The accuracy of our extraction rules in the development, resp. test set and obtain is 71.35%, respectively and 69.63%.

Evaluation of the end-to-end system We selected randomly 38 sentences for the test set (from which 26 were considered as extracted correctly at previous step – section 3). We gave the corresponding generated queries $q = M + L + T$ as input to the INSEE-Search system [3]. We evaluated the accuracy of the system using a modified version of the mean average precision metric, (MAP) widely used for evaluating ranked lists of results. MAP is traditionally defined based on a binary relevance judgment (relevant or irrelevant in our case). We experimented with the two possibilities:

- MAP_h is the mean average precision where only highly relevant datasets are considered as relevant ($MAP_h(10)$ is computed on the top 10 search results).
- MAP_p is the mean average precision where both partially and highly relevant datasets are considered relevant.

Note that there is no guarantee that any "highly relevant" element at all exists in the dataset for each query.

The results (Table 2) show that, given an arbitrary claim (related to statistic entities), fine-grained and relevant information can be returned in the vast majority of the cases. They also show that, as in all keyword-based search systems, building a perfect query is neither necessary or sufficient for obtaining good results. Even if a good entity extraction improves the results, we can still find highly or partially relevant information even if the entity extraction is not perfectly achieved. Our findings should be confirmed by an evaluation on more claims, more databases and in a real-user study. We also showed in [3] that the performance of our query system was similar to a document-level search engine such as Google, but with a much better granularity (data cell instead of page).

4 Related Work and Perspectives

BONIE [8] claims to be the first open numerical relation extractor. The system is based on high precision patterns to extract seed facts from input sentences

[9] *"taux de chômage", "nombre de demandeurs d'emploi", "niveau de vie", "consommation des ménages", "PIB", "inflation", "SMIC", "taux d'emploi"*

[10] The annotated data is available at https://gitlab.inria.fr/tcao/text2insee/

	$MAP_h(10)$	$MAP_p(10)$
Overall performance (38 sentences)	**0.672**	**0.789**
among which M extracted correctly (26)	0.725	0.829
M extracted incorrectly (12)	0.559	0.703

Table 2. Evaluation of INSEE-Search.

and on bootstrapping to increase the number of seed facts and to learn patterns. We tried their approach, but found that the learnedt patterns were either too generic or too specific and failed to capture the correct dependency path in the new texts. ClausIE [5] is an open information extraction system. It first detects clauses in a sentence and then apply specific rules for each type of clause in order to extract the entity of interest. ClausIE also makes use of a hand-crafted dictionary of verbs to identify the existence of relation in sentence. Compare to their approach, we have a "semi-automated" solution to identify the list of verbs. ClaimBuster [6] was the first work on check-worthiness. They used annotated sentences from US election debates to train a SVM classifier in order to determine whether or not a sentence is a check-worthy claim. This is the common approach when having a large amount of training data, which is not the case in French.

In this article we have presented an end-to-end system for identifying statistic claims and finding in a statistic database the relevant statistic data for checking this claim. A classic defect of these pipeline approaches in NLP systems is that errors accumulate at each step. Nevertheless, our results show that we often manage to find useful information for the user, which will make the human work of fact-checking easier and faster. To make the RDF graph up-to-date, our crawler works on a daily basis to collect the latest statistic tables. We also leave journalists state whether the claim is "true", "mostly true", "mostly false" etc.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (Mar 2003)
2. Cao, T., Manolescu, I., Tannier, X.: Extracting linked data from statistic spreadsheets. In: Int'l. Workshop on Semantic Big Data (2017)
3. Cao, T.D., Manolescu, I., Tannier, X.: Searching for Truth in a Database of Statistics. In: WebDB (2018)
4. Cazalens, S., Lamarre, P., Leblay, J., Manolescu, I., Tannier, X.: A content management perspective on fact-checking. In: WWW (2018)
5. Corro, L.D., Gemulla, R.: ClausIE : Clause-Based Open Information Extraction. In: WWW (2013)
6. Hassan, N., Zhang, G., Arslan, F., Caraballo, J., Jimenez, D., Gawsane, S., Hasan, S., Joseph, M., Kulkarni, A., Nayak, A.K., Sable, V., Li, C., Tremayne, M.: Claimbuster: The first-ever end-to-end fact-checking system. PVLDB (2017)
7. Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: EMNLP (2015)
8. Saha, S., Pal, H., Mausam: Bootstrapping for Numerical Open IE. ACL (2017)
9. Strötgen, J., Gertz, M.: HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In: Int'l. Workshop on Semantic Evaluation (2010)