Implementing Eco's Model Reader with Word Embeddings. An Experiment on Facebook Ideological Bots

Leonardo Sanna, Dario Compagno

Implementing Eco's Model Reader with Word Embeddings. An Experiment on Facebook Ideological Bots

Leonardo Sanna[1], Dario Compagno [2]

[1] Università di Modena e Reggio Emilia – leonardo.sanna@unimore.it

[2] CREM – Centre de Recherche sur les médiations – Université de Lorraine – dario.compagno@univ-lorraine.fr

Abstract

In semiotics, the concept of model reader is used to describe the felicity conditions of a text, that is, the information and pragmatic competence needed to interpret the text with reference to a hypothesis on its producer's intention. The model reader permits to formulate inferences about the implicit content of sentences and of the entire text. In this paper we propose to formalize the model reader as a function that takes as inputs a text and a larger context and produces as output what is needed to complete the text's implicit information, filling up its "blank spaces". One possible technique to implement this function is word embedding. We performed an experiment in this sense, using the data collected and analyzed by the Tracking Exposed group (TREX) during the Italian 2018 elections. For their study, six blank Facebook profiles were created, each characterized by a political orientation: all profiles followed the same common set of 30 pages, representative of the entire Italian political spectrum at the time, but each profile interacted only with content linked to its distinctive political orientation. TREX's study of the profiles' newsfeeds demonstrated that each profile was prompted with an uneven distribution of information sources, biased by its political orientation. For our study, we created six different word spaces, one for each profile. Then we identified a certain number of politically neutral terms and observed the semantic associations of these terms in each word space. To identify the terms, we performed a classification of the entire corpus with the software Iramuteq and selected the most significant terms associated with each cluster. Finally, by performing some operations within each word space, we observed some differences in semantic associations that are coherent with the political orientation of the corresponding profile. These results appear to show that word embedding is a valuable approach for computational text pragmatics, as they can help to model the inferences performed by a certain reader. Also, these results suggest the pertinence of such analyses for the study of filter bubbles resulting from algorithmic personalization.

Keywords: semiotics, word embedding, social media, filter bubble, machine learning

1. Implementing the concept of model reader

The concept of model reader was created by Umberto Eco (1979) to represent the pragmatic competence needed to interpret a certain text with reference to its producer's intention or, better, to a given hypothesis about this intention. We could think of the model reader as a set of implicit instructions to formulate legitimate inferences from sentences and their combinations in texts. One of the main components of the model reader is what Eco called encyclopedic competence, accessing a cultural context shared by the text's producer and reader. For instance, to understand the sentence "Macron nomme Philippe" [Macron nominates Philippe], the reader has to know that Macron is the actual French President, that the President has the power to nominate a Prime Minister, and more importantly the reader has to guess also (by abduction) that this sentence refers to such nomination. In Eco's theory, these steps are accounted for by the competence of a model reader adequate to that sentence, including knowledge about the actual world, it's entities and their interactions. In absence of such encyclopedic interpretation, it would be impossible for the reader even to guess the

correct meaning of the verb "nommer" in this context. This competence has to be acquired by a previous knowledge of other texts and verbal exchanges. For example, to fully understand the last chapter of the Star Wars movie franchise, I am required to know the previous episodes, since the plot has plenty of intertextual references pointing to them. Hence, we should intend the encyclopedic competence of the model reader as a shared cultural background, that in many cases might be expressed by intertextuality.

In order to treat the concept of model reader empirically, we propose a tentative formalization of it. We argue that it can be seen as an inferential model, produced as output from a function that takes as input a target text and a larger corpus of texts. This model should then be able to add to the target text the implicit information needed for its interpretation. Reading, as an activity, therefore is accounted for by two circular steps: the identification of some reading instructions from a text and their application to the text itself. If such formalization of the model reader could be implemented computationally, it would make it easier to treat textual production and interpretation automatically. We also argue that it would become possible to perform experiments so to observe whether or not a model reader, and the textual interpretation deriving from its application, is affected by algorithmic personalization.

Since 2011, the hypothesis of filter bubbles (Parisier 2011) has pointed out that in an era of algorithmic personalization, there is a danger of not sharing any common ground anymore, each of us living in his or her personal information bubble[1]. This is because each person has access to a richer variety of sources compared to the pre-Internet era, and so people make use of algorithms to retrieve the information that is pertinent to them. In this work, we are interested in understanding whether (and to what extent) the model reader of a set of texts is influenced by filter bubbles. In other words, how the access to information affects textual interpretation, by constraining the constitution of a largely shared common knowledge. We point out that this research goal is particularly complex since the study of algorithmic personalization has in itself some methodological issues that are still unsolved (i.e., how to control for all the variables of users behavior that may influence the filtering algorithm, such as user activity on and outside social media platforms). Another important factor to take into consideration is that a large amount of digital data has to be collected and analyzed to study algorithmic personalization. As a first step towards accomplishing this task, we used word embedding to implement the model reader of some text fluxes.

2. Word embedding

Word embedding is a machine learning technique used in natural language processing to create semantic vector space models, in which every word in the model is mapped to a vector so that words with similar meanings are represented as close to one another. The distributional hypothesis lies behind this approach (Harris 1954), affirming that words with similar meanings usually occur in similar contexts, and so that some semantic insight can be derived from the analysis of numerous contexts. This concept can be expressed in mathematical terms by saying that "The degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear" (Lenci 2008).

Nowadays several algorithms implementing word embedding exist for example *word2vec* (Mikolov et al. 2013a), *fastText* (Bojanowski et al. 2017), *Star Space* (Wu et al. 2018) and

---

[1] The hypothesis of filter bubbles does not make a universal scientific consensus. Some studies show that individuals may have access to a large shared background (Compagno et al. 2017, Bechmann 2018).

RAND-WALK (Arora et al 2015). In this work we used *word2vec*, because it is currently the most popular and best documented. The software implements two different neural networks algorithms, the one being the inverse of the other (Mikolov et al. 2013b): continuous-bag-of-words models (CBOW) and skip-gram models. A CBOW model is used to predict the probable occurrence of a target word in some given context. Instead, a skip-gram model works by predicting a probable context starting from a target word (Figure 1). In a nutshell, what word embedding does is to create a representation of a corpus based on the probability of co-occurrence of words in contexts. This means that it can also capture semantic relations that are *in absentia*, i.e. between words that never actually co-occur in the corpus whilst having a similar meaning. For this reason, among others, word embeddings are not simply calculations of word occurrences, and may better simulate inferential models (see below, § 4).
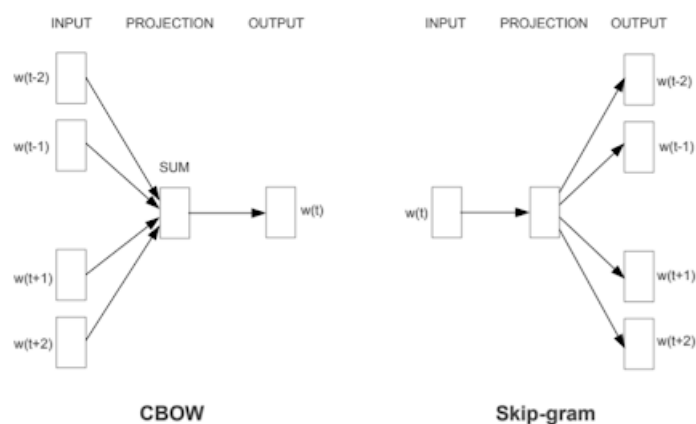

Figure 1: the functioning of word2vec, from Mikolov et al. (2013b)

Word2Vec can work either with negative sampling or with hierarchical softmax as explained in the paper by Mikolov et al. (2013a). We are using Word2Vec in the version implemented in Python by Gensim (Rehurek and Sojka 2010), that uses negative sampling by default[2].

3. Dataset

The dataset on which we conducted our experiment was collected by the Tracking Exposed (TREX) research group[3] during the Italian elections in 2018[4]. The TREX group has a main goal to highlight (expose) user profiling and algorithmic personalization. For their 2018 experiment (Hargreaves et al. 2018a), they built six virgin Facebook profiles (henceforth "bots") to do some automatic data acquisition. Each profile automatically scrolled on his feed 30 posts per hour, collecting and parsing HTML code with a browser extension developed by the research team[5]. The extension was used to collect evidence about what was being shown to the bot.

To control for the algorithmic selection of the FB algorithm, all bots followed the same 30 different sources, covering the Italian political spectrum of that time (see Hargreaves et al. 2018b for details). However, each bot was interacting (by liking) only with the content

---

[2] For reference see http://bit.ly/word2vec-docs.

[3] Official website https://tracking.exposed/.

[4] Database available on Github at https://github.com/tracking-exposed/experiments-data.

[5] More information on the extension here https://facebook.tracking.exposed/.

coming from one particular political segment[6]. Therefore, we had a far-right bot, a right-wing bot, a left-wing bot, a far-left bot, a populist bot and finally an undecided bot that was not interacting with any content in its feed. The dataset is hence composed of two main parts: the *sources* dataset, including the totality of posts made by the 30 pages (between January 10th and March 6th, 2018) and the *impressions* dataset, including only the posts that have been shown to the bots. In their 2018 study, the TREX researchers found evidence of uneven exposure to the sources of information, discovering that usually exposure was unbalanced towards the political bias of the bots. Starting from the same pages, the six bots accessed six different feeds (Figure 2: the x-axis shows the first 40 posts on the bot's feed, the y-axis the percentage of them according to their political connotation).
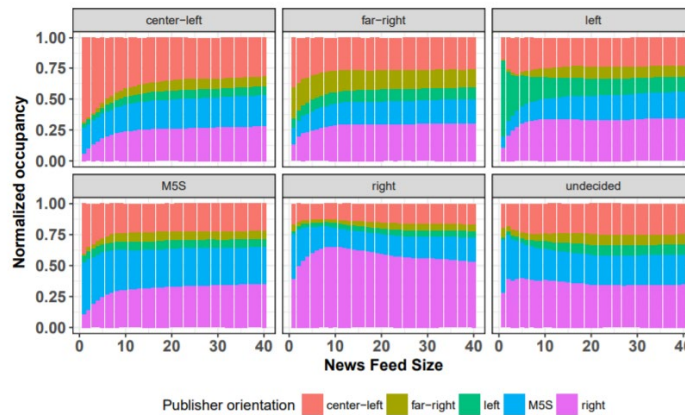

Figure 2: Uneven distribution on Facebook feeds, from Hargreaves et al. (2018a),
reproduced with permission

Now, if we consider algorithmic personalization from the perspective of Umberto Eco's semiotics, the politically biased feeds of the six bots contribute to the shaping of six different model readers. In other words, the textual flux shown to a FB account influences the underlying instructions that guide the reading of the flux itself: for example, a person who gets his or her information mainly from far-right sources should realize certain inferences more easily than another person reading far-left sources, and vice versa. Because of this, we expect to find six different inferential models in the TREX dataset, one for each bot. Our aim is to simulate these six inferential models by training six different word embedding spaces and studying their behavior.

4. Experiment

First of all, it is important to highlight that the inferential model we implement with word embedding may perform two different kinds of inferences:

1. *Necessary inferences* are all those inferences driven by common word co-occurrences in the corpus. These inferences are necessary for a basic understanding of the semantic content of the text. They include *mandatory inferences* (i.e. compound words) and

---

[6] Each segment includes pages associated with political parties, personalities, newspapers. The main parties associated with each segment are the following. Left: Possibile, Sinistra Italiana; center-left: Partito Democratico; right: Lega Nord, Fratelli d'Italia; far-right: Forza Nuova, Casapound; populist: Movimento 5 stelle.

natural inferences such as technical lexicon (names of people, institutions, laws). For instance, "step" and "out" –> "step out" and "Macron" –> "President" are examples of necessary inferences.

2. *Embedded inferences* are all those inferences made without starting from direct co-occurrences of words in the corpus. The capacity to detect these inferences is the real peculiarity of word embedding. These inferences are interesting because they provide interpretation paths that would not be easily discovered by standard semiotic analysis or corpus linguistics. For example, "Leader" + "Germany" –> "Merkel" is an example of embedded inference.

To study the behavior of the inferential models simulated by word embedding, we wanted to identify some words that were used all along the entire Italian political spectrum. Starting from these words we could then observe which inferences they generated in different model readers (associated to our six bots). We first used Iramuteq[7] to perform a hierarchical classification (Reinert 1983) on the sources dataset: we identified the main topics in the entire debate and the words characterizing these topics, independently from the eventual political filtering operated by the FB algorithm. Figure 3 shows a correspondence analysis displaying the 11 clusters and the most specific words for each, that is, the words which are most associated with a cluster and less with all the others.
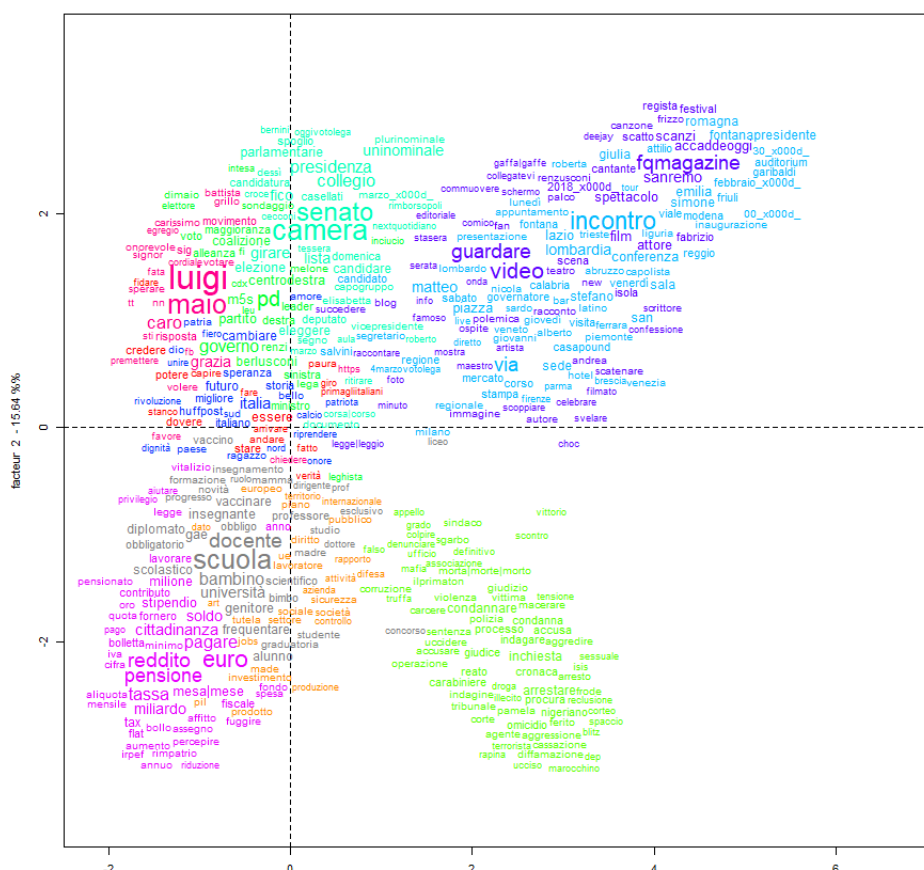

Figure 3: Correspondence analysis of the clusters in the sources corpus

Class 1 (14,7%): "Verb Modalities"	Class 2 (12,39%): "Occupation"	Class 3 (14,52%) "Crime News"	Class 4 (10,78%) "Taxes & Pensions"
1. essere/stare	1. lavoratore	1. arrestare	1. euro
2. credere	2. settore	2. indagare	2. reddito
3. potere	3. europe	3. inchiesta	3. pensione
4. dovere	4. investimento	4. cronaca	4. tassa
5. pensare	5. sociale	5. nigeriano	5. pagare
6. vedere	6. tutela	6. condanna	6. cittadinanza
7. capire	7. assunzione	7. procura	7. soldo
8. andare	8. diritto	8. sentenza	8. stipendio
9. paura	9. sicurezza	9. omicidio	9. "flat tax"
10. governare	10. made in Italy	10. processo	10. fornero
11. sentire	11. economia	11. reato	11. lavorare
12. fare	12. prodotto	12. giudice	12. fiscale
13. promettere	13. salute	13. carabiniere	13. bolletta
14. popolo	14. ue	14. accusa	14. disoccupazione
15. verità	15. occupazione	15. polizia	15. versare

Table 1: Main clusters composition

After having identified these 60 frequent and supposedly neutral words, we performed word embedding on the impressions dataset, creating one model for each bot. Then, by using the lists of words obtained with our classification above, we explored our models so to see how our bots "make inferences": which new words does each bot associate to the 60 extracted from the four clusters?

We created an embedding for each bot, building both a skip-gram model and a C-BOW model, using in both cases a window context of 10 words and 200 dimensions. We used the plain text of each post to create our models[8]. In Table 2 we summarize the size of each model.

	Center-Left	Far-Right	Left	Populist	Right	Undecided
Word Types	12615	13712	18326	24930	25893	18945
Word Tokens	89600	107254	180629	312954	376813	187033
Posts	3694	3417	6437	13331	16085	7597

Table 2: The size of the different sub-corpora for each embedding. Duplicated posts have been removed.

---

[8] For each post in the impressions dataset we have: the date of publication, the date of impression, the number of comments, the name of the publisher, impression order (the position in the newsfeed at the time of the impression), the permaLink of the post, the URL of the post (sometimes external), the post ID, the publisher orientation, the bot political orientation, the number of visualizations. We filtered our posts by political orientation, mapping the post ID within the sources dataset in which we also have the "post message" available. The post message is the plain text of each Facebook post. For our goals we only need the post message of each impression.

Each sub-corpus has been preprocessed removing URLs, emojis and terms shorter than 2 characters; each word was also turned to lowercase during the tokenization process. Word2vec works by default with a downsampling[9] defined by Formula 1.

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

Formula 1: Downsampling for word embedding

This basically ignores the most frequent words (like articles and prepositions), that do not add any semantic information to our model. However, in our corpus we obtained better results without downsampling[10]. This is probably caused by the relatively small size of our dataset; word embedding algorithms are designed to work with very large datasets and the default downsampling threshold has been determined heuristically, as stated in the original paper (Mikolov et al. 2013a).

For each of the four clusters, we computed the cosine similarity of the top-20 most similar words. The results are summarized below (Figures 4-7). For visualization we selected samples of words including of:

- The two most similar adjectives
- The two most similar nouns
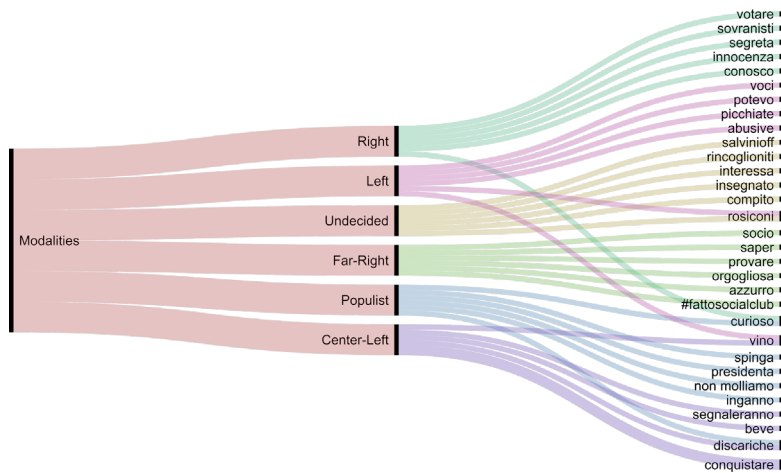- The two most similar verbs

Figure 4: Inferences within the "Modalities" cluster

---

[9] According to Mikolov et al. (2013a) "each word w_i in the training set is discarded with probability computed by the formula (1), where $f(w_i)$ is the frequency of word w_i and t is a chosen threshold, typically around 10^{-5}."

[10] This is part of the Gensim implementation that recommends using a threshold in a range between 0 and 10-5. In our work we determined, heuristically, that the best value for us was 0.
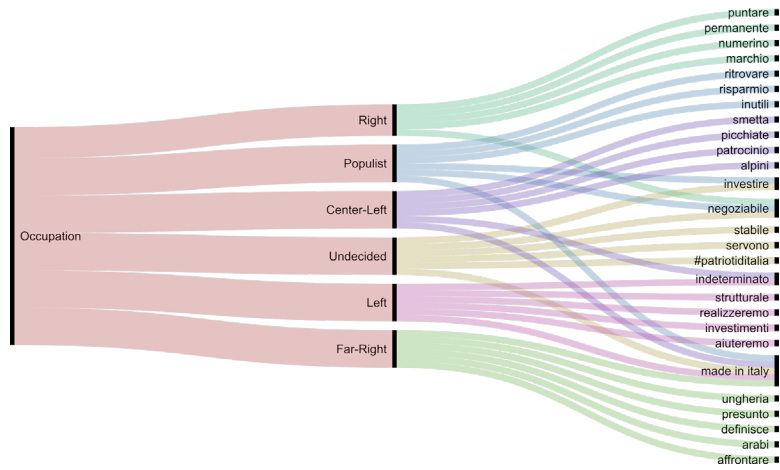
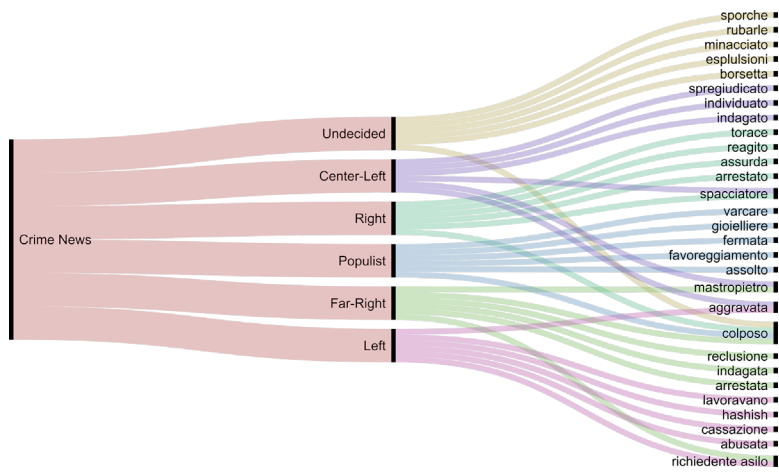*Figure 5 : Inferences within the "Occupation" cluster*

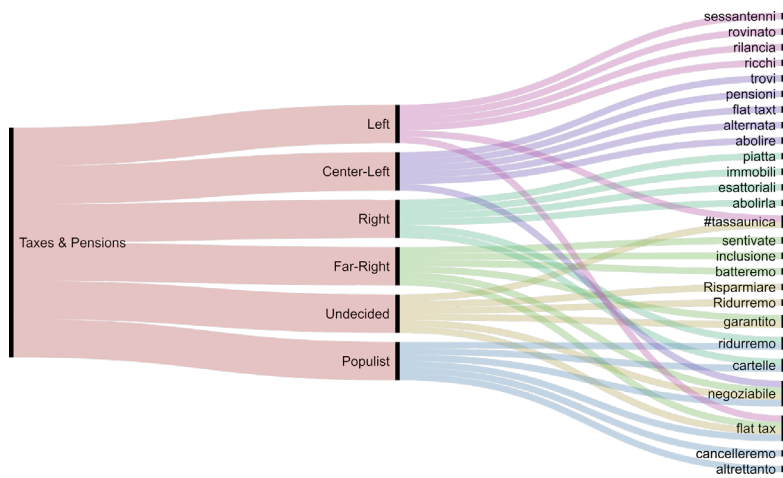*Figure 6 : Inferences within the "Crime News" cluster*

Figure 7: Inferences within the "Taxes and Pensions" cluster

This first level of inferences portrays a situation in which there is a lot of lexical diversity, suggesting that we can really see different inferential models in action. For example, each bot associated very specific words to the *Occupation* cluster, not shared by the other bots, so reframing the problematic from a unique perspective. There is an exception: all bots seem to recall the expression "Made in Italy" when making inferences in the occupation domain. However, this does not mean that the inferred words necessarily have the same connotations for all bots: in the *Crime News* cluster, the expression "richiedente asilo" (asylum seeker) is inferred both by the Left and the Far-right bots, but probably a deeper analysis would show two contrasting axiological positionings towards the expression. Still, by qualitative analysis of the inferred terms, we observed that those used by right-wing parties seem dominant in the global debate. In the two economic clusters, the expression "flat tax" and "made in Italy" are shared by many bots, and both expressions were crucial in the right-wing parties electoral discourse. As first partial result it seems, then, that if different bots can be associated to different model readers, each with its own perspective, they are also influenced by the global information as a whole.

We can try to go more into the details of the inferential models by calculating *snowball inferences* (Rogers 2017) to compare different interpretation paths. For brevity of exposition, we focus here on one single case. We saw that "flat tax" is a term appearing in many inferential models, even if it originates from the electoral discourse of Italian right. Its mention alone therefore does not say much about how different model readers may give a value to it. Hence, we tried to unpack the semantic frame (Fillmore 1976, Eco 1979) of the expression "flat tax". To prevent potential biases in our findings, we started by taking a step back and working on the more generic word "tassa" (tax). This allows us to observe a larger frame about taxes in our corpus. Figure 8 displays the indirect inferences realized by the politically undecided bot.
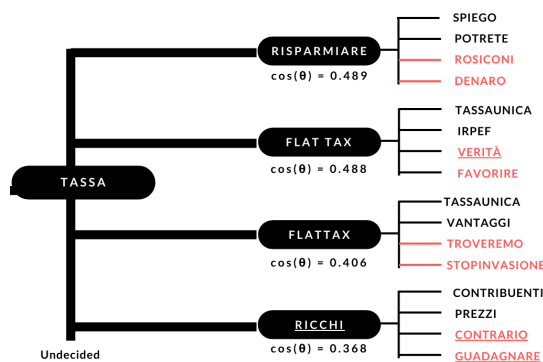

Figure 8: Examples of inferential paths made by the politically undecided bot

We analyzed qualitatively the resulting spaces of word embedding to identify the positive or negative axiological perspectives given to the main word vectors. We also manually distinguished embedded and necessary inferences (see §4 above) by analyzing collocations. In Figures 8 and 9 we show the embedded inferences in red and the necessary ones in black. We selected four out of the ten most similar words, including at least two embedded inferences for each. This allows us to visualize and possibly better understand the ideological differences of the six bots. The underlined words evoke negative axiological frames in the word embedding model of this bot.

In the case of Figure 8, showing data from the politically undecided bot, three out of four of the main inferred words are positive or neutral towards the term "flat tax". If we expand further the frame of each sub-word, it becomes clearer that positive evaluations are predominant and that the ideology proposed to this bot is close to that of the Italian right. For

instance, the word "verità" (truth) is associated with a semantic frame of mistrust and falsehood. This method allows for manual comparisons among different inferential models, as it can be shown by visualising the inferences induced by the same term "tassa"(tax) in the far-right bot (Figure 9).
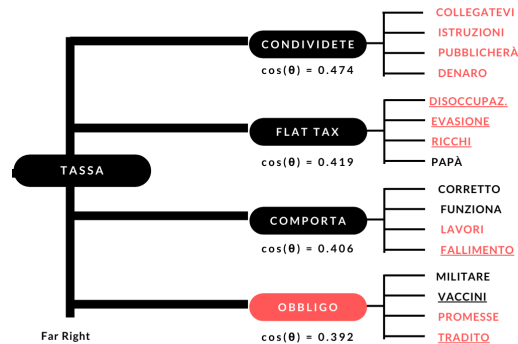

Figure 7: Examples of inferential paths made by the far-right bot

In Figure 9 we observe a different axiological configuration compared to Figure 8, as "flat tax" leads to embedded inferences such as "disoccupazione" (unemployment), "evasione" (tax evasion) and "ricchi" (the rich), inducing a negative evaluation of introducing a flat taxing system in Italy, as it would become an advantage only for the rich. We reiterate that the inferential models that we can extrapolate with word embedding are complex to interpret, and for now they can only be used as tools for supporting quali-quantitative analysis.

5. Discussion

With our study we distinguished how different bots infer some given words from others, depending on the information they are shown by the FB algorithm. Despite the expected differences among them, we also found evidence that, during the 2018 campaign, in Italy, the right-wing arguments were dominant across all different bots, showing that algorithmic personalization may actually propose to readers information in contrast to their individual political perspective and oriented by more global tendencies. If confirmed, these results would be important as it would show that Facebook influences the public discourse as it performs a sort of automated agenda setting skewed towards the most popular perspectives.

It may also be that what we observed was an artefact due to our choice of using word embedding. In this case it would still be interesting to understand why right-wind information managed to affect the construction of all the other vector spaces. This may be due to two factors: repetition and corpus size. First, in the right-wing sub-corpus there are a lot of reposts and more generally repetitive content. Words that occur a lot may then be overrepresented, and the interpretation of more rare words risks to be problematic. However, the word "flat tax" despite being associated with the Italian right-wing discourse, does not appear in our top-20 most similar words list of the inferences of the right-wing bot. This might be caused by the size of the right-wing sub-corpus, which was the largest; hence the probability of occurrence of the expression "flat tax" is smaller compared to the other dataset. It is anyway evident that Facebook algorithms spread right-wing content more, since "flat tax" is found in the models of all other bots.

Despite the fact that the sub-corpora are of different size, we managed to compute acceptable semantic models, capturing meaningful semantic relations for every bot. However, the quality among models is variable. An effect of corpus size may explain why the term "vino" [wine] appears in one of the models, apparently without any reason. We also observed what was suggested by Rogers (2018): the use of ideologically charged terms differentiates vocabularies across bots, making it difficult to compare them with purely computational methods.

As this is a first attempt to formalize the concept of model reader so to simulate it with word embeddings, we do not have enough data to validate the effectiveness of our simulation. Anyway, literature is still quite uncertain about evaluation methods for word embeddings (Mikolov et al. 2013b, Gimenez et al. 2015, Lai et al. 2016, Sahlgren and Lenci 2016, Naili et al. 2017, Wang et al. 2019). In general, for our ends, we did not notice any relevant difference between the use of skip-gram or C-BOW models, nor between negative sampling and hierarchical softmax. Hierarchical softmax could allow us to control better for repetitive content, but to validate this assertion a dedicated work is needed.

6. Conclusion

This experiment showed that Eco's model reader (1979) might be formalized as an inferential model. In principle, this means that it can be simulated with word embedding to create computational inferential models, that can be built empirically and compared so to identify a possible divergence of interpretation paths. Our experiment conducted on Facebook showed that algorithmic personalization seems to impact on model readers, differentiating the most relevant lexicon of different text fluxes, but further research is needed to confirm this statement.

We obtained a confirmation that political sources are treated unevenly by the Facebook algorithm; particularly, we collected evidence that in 2018 Italy, right-wing vocabulary had spread better than the others. We suggest that further research should focus on the broad question, whether or not word embedding spaces actually correspond to the semantics of the texts used to build them. This task may demand for evaluation performed by qualitative exploration.

References

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2015). Rand-walk: A latent variable model approach to word embeddings. *arXiv preprint*, arXiv:1502.03520.

Bechmann, A. (2018). Filter Bubbles: From Academic Debate to Robust Empirical Analysis. *International Communication Association Conference*, May 2018, Prague.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135-146.

Compagno, D., Mercier, A., Mesangeau, J. & Chelghoum, K., (2017) La reconfiguration de l'offre d'information médiatique opérée par les réseaux socionumériques. *Réseaux*, 205:91-116.

Eco, U. (1979) *The Role of the Reader*, Bloomington : Indiana UP.

Fillmore, C. J. (1976) Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20-32.

Hargreaves, E., Agosti C., Menasché D., Neglia G., Reiffers-Masson A., Altman E (2018a) Biases in the Facebook News Feed: a Case Study on the Italian Elections. *International Conference on Advances in Social Networks Analysis and Mining*, August 2018, Barcelona

Hargreaves, E., Agosti C., Menasché, D., Neglia, G., Reiffers-Masson, A., Altman, E. (2018b) Supplementary Material for Biases In The Facebook News Feed: A Case Study On The Italian Elections. *International Symposium on Foundations of Open Source Intelligence and Security Informatics*, August 2018, Barcelona.

Harris, Z.S. (1954) Distributional structure. *WORD*, 10:2-3, 146-162.

Lai, S., Liu, K., Xu, L., Zhao, J. (2016) How to generate a good word embedding. *IEEE Intelligent Systems,* 31.6:5-14.

Lenci, A. (2008) Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1-31.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013a) Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, Dicembre 2013, Leake Tahoe, 3111-3119.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013b) Efficient estimation of word representations in vector space. *arXiv preprint*, arXiv:1301.3781.

Minarro-Giménez, J.A., Oscar M.-A., & Matthias Samwald. (2015) Applying deep learning techniques on medical corpora from the world wide web: a prototypical system and evaluation. *arXiv preprint*, arXiv:1502.03682 .

Naili, M., Habacha Chaibi, A. and Hajjami Ben Ghezala, H. (2017) Comparative study of word embedding methods in topic segmentation. *Procedia computer science*, 112 : 340-349.

Parisier, E. (2011) "The Filter Bubble: What the Internet is hiding from you" The Penguin Press - New York

Rehurek, Radim, and Petr Sojka.(2010) "Software framework for topic modelling with large corpora." In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

Reinert, A. "Une méthode de classification descendante hiérarchique: application à l'analyse lexicale par contexte." (1983) Cahiers de l'Analyse des Données 8.2 : 187-198.

Rogers, Richard. "Foundations of digital methods: Query design." (2017): in The Datafied Society Studying Culture through Data 75-94 - Amsterdam University Press.

Rogers, Richard. (2018) "Otherwise engaged: Social media from vanity metrics to critical analytics." International Journal of Communication 12.732942 : 450-72.

Sahlgren, Magnus, and Alessandro Lenci (2016). "The effects of data size and frequency range on distributional semantic models." arXiv preprint arXiv:1609.08293

Wang, Bin, et al. "Evaluating word embedding models: methods and experimental results." (2019) APSIPA Transactions on Signal and Information Processing 8 .

Wu, L. Y., Fisch, A., Chopra, S., Adams, K., Bordes, A., & Weston, J. (2018). Starspace: Embed all the things!. In Thirty-Second AAAI Conference on Artificial Intelligence.