LSACoNet: A Combination of Lexical and Conceptual Features for Analysis of Fake News Spreaders on Twitter
Notebook for PAN at CLEF 2020

Hamed Babaei Giglou[1], Jafar Razmara[1], Mostafa Rahgouy[2], and Mahsa Sanaei[1]

[1]Department of Computer Science, University of Tabriz, Tabriz, Iran
h.babaei98@ms.tabrizu.ac.ir, razmara@tabrizu.ac.ir,
mahsasanaei97@ms.tabrizu.ac.ir
[2]Part AI Research Center, Tehran, Iran
mostafa.rahgouy@partdp.ai

Abstract Fake news detection on social medial has attracted a huge body of research as one of the most important tasks of social analysis in recent years. In this task, given a Twitter feed, the goal is to identify fake/real news authors or spreaders. We assume fake news authors mostly like to play with the semantic aspect of news rather than trying to add specific changes to their styles. However, making a change into the semantic aspect of news can cause unwanted changes in style. We hypothesize, by relying on news content, a combination of semantic and coarse-grained features may lead us to common information about the author's style while reviewing the conceptual aspect of author documents. In this paper, we propose the LSACoNet representation using a fully connected neural network (FCNN) classifier that combines different levels of document representation to investigate this hypothesis. Experimental results presented in this paper showed that a combination of representations plays an important role in identifying fake/real news spreaders. Finally, we achieved accuracies of 72.5% and 74.5% in the English and Spanish test datasets, respectively, using presented LSACoNet representation and FCNN classifier.

Keywords: Fake News, False Information, Feature Combination, Suspicious Fake News Authors, Fully Connected Neural Network

1 Introduction

False information such as fake news is one of the main threats of our society. In the last years, big social networks like Facebook or Twitter have admitted that their networks had fake and duplicate accounts. Regarding this, fake news are not a new phenomenon, and the exponential growth of social media has offered an easy way for fast propagation. These fake news usually try to deceive users to express specific options. Users play a

critical role in the creation and spread of fake news by influencing people to make a decision, support or attack an idea, or even election candidate.

This year at author profiling tasks series, the new task got a place to convey our concern to stop spreading fake news, a Profiling Fake News Spreaders on Twitter [16] task. In this task, we aim to identify possible fake news spreaders on social media as a first step towards preventing fake news from being propagated among online users.

> Task: Given a Twitter feed, determine whether its author is keen to be a spreader of fake news.

The main goal was aimed to investigating if it is possible to discriminate authors that have shared some fake news in the past from those that, to the best of our knowledge, have never done it. Also, this task runs based on a multilingual perspective for English and Spanish languages.

The rest of the paper is organized as follows. Section 2 presents related works. Section 3 describes the proposed method. Section 4 describes the performed baselines, experiments, and discusses the obtained results. Finally, section 5 presents our conclusions.

2   Related Work

In the fake news challenge (FNC-1) [6] shared task, studies have been done with 50 participating teams. They performed a detailed feature analysis of participant and concluded that identifying high-performing features for the task yields a new model which mostly rely on the lexical overlap for classification. They believe that this task is challenging since the best performing features are not yet able to resolve difficult cases. Thus, more sophisticated machine learning techniques are needed, which have a deeper semantic understanding. In [18], the authors made a study to understand user profiles on social media for fake news detection and proposed a principled way to understand which features of user-profiles are helpful for fake news detection. They concluded that first, there are specific users who are more likely to trust fake news than real news, second, these users reveal different features from those who are more likely to trust real news. These observations showed the importance of feature construction for fake news detection.

According to a major study of [6] and the study of [18], we believe, this task is sensitive to feature dimensionality. That is, low-quality features can reduce overall model performance. Feature combination is one of the common actions used to enhance features. In combination methods, different feature vectors are lumped into a single long composite vector, or in addition to the combination of feature vectors, the dimension of feature space is reduced. From an NLP attitude, many methods proposed to employ feature combinations for different studies like fake news challenge.

A work done in [21] has studied false information on Twitter. They found that real tweets contain fewer bias markers, hedges, subjective terms, and less harmful words. They build a model that combined features like graph-based, cues words, and syntax. They concluded, incorporating linguistic features and social network interactions with neural network models improves the classification of suspicious news. However, they

are expecting to utilize more sophisticated discourse and pragmatics features and inferring degrees of credibility in their future works.

In the work of [7], they have used a Long Short-Term Memory (LSTM) network combined with other features such as bag-of-characters (BOC), BOW, and topic model features based on non-negative matrix factorization, Latent Dirichlet Allocation, and Latent Semantic Indexing. They achieved a state-of-the-art result of 60.9% (Macro F1) on the Fake News Challenge (FNC-1) dataset. Similar to this work, at [4], an approach was presented that combines lexical, word embeddings, and n-gram features to detect the stance in fake news. Their approach has been tested on the FNC-1 dataset and achieved an accuracy of 59.6% (Macro F1) close to state-of-the-art results using a simple feature representation. Mainly approaches at Fake News Challenge (FNC-1) dataset incorporated a different combination of features, such as word or character n-grams, bag-of-words, word embeddings, latent semantic analysis features [17] [8].

At another work [13], they have used a set of linguistic features like n-grams, punctuation, psycholinguistic, readability, and syntax features. The proposed linguistics-driven approach suggests that to differentiate between fake and genuine content it is worthwhile to look at the lexical, syntactic, and semantic level of a news item in question. They have achieved an accuracy of up to 76% in their own collected dataset.

3 Proposed Approach

We assume authors may convey different concepts when they are tweeting, so differences in concepts can capture fake/real news. Since fake news spreaders can be very smart or complicate their semantic of tweet concepts highly keen to be real but in a different style than usual. According to [15] coarse-grained features are most likely to find author's styles. So, taking author fingerprinted features into account can be useful in the case of finding author styles. To construct a hypothesis, lets (X^i, y^i) be the definition of each user tweets. X^i refers to user i tweets. y^i describes fake/real news spreader. Suppose $i \in [1, m], j \in [1, n]$ and m, n be the maximum numbers of users, and each user tweets, respectively. We can define $X^i = \cup_{j=1}^{n} \gamma_j$ in which γ_j refers to array of words which belongs to j-th tweet's for user i-th, and γ_j^k is the k-th word of array γ_j with length of $|\gamma_j|$. In the following, we will use these notations to introduce our proposed approach in more details.

3.1 Data Preprocessing

In the first stage of preprocessing, we used Preprocessor[1] which is a preprocessing library for tweet data written in Python. It used to remove URLs, Hashtags, Mentions, Reserved words (RT, FAV), Emojis, Smileys, and Numbers from X^i even those that already masked in the dataset. Next, punctuation removal, stopwords removal, and stemming applied to $\forall \gamma_j^k$, $k \in |\gamma_j|$ using NLTK 3.0 Toolkit [1].

---

[1] https://github.com/s/preprocessor

3.2 Data Representation Methods

I. ConceptNet Numberbatch Regarding word embeddings that represent only distributional semantics like Word2Vec or GloVe and word embedding that represent only relational knowledge like ConceptNet, ConceptNet Numberbatch is a hybrid word embedding built using an ensemble approach. It combines data from ConceptNet, Word2Vec, GloVe, and OpenSubtitles 2016 using a variation on retrofitting [19].

- **ConceptNet** [19] is a knowledge graph that connects words and phrases of natural language with labeled edges. ConceptNet sources include symmetric and asymmetric relations. Its knowledge is collected from many sources that include expert-created resources, crowd-sourcing, and games with a purpose. It is designed to allow the applications to better understand the meanings behind the words people use [19].
- **GloVe** [12] is a vector space with meaningful substructure which pre-trained on various datasets.
- **Word2Vec** [11] is a word vectors pre-trained on the Google News dataset.
- **OpenSubtitles 2016** [20] is a collection of movie subtitles and used as a part of meta data for training ConceptNet Numberbatch.

ConceptNet Numberbatch is a multilingual word embedding and represents 78 different languages in 300 dimensions. Words in different languages share a common semantic space, and that semantic space is informed by all of the languages. The f is a representation of semantic space.

$$f : Word \longmapsto V_{300}^1$$

In this work, we used ConceptNet Numberbatch version 19.08, and a vocabulary size of 651859 for Spanish, and 516782 for English. Γ uses f to represent word vectors for both words in numberbatch vocabulary and OOV words.

$$\Gamma(word) = \begin{cases} \overrightarrow{f}(word), & \text{word} \in f \\ \overrightarrow{0}, & \text{word} \notin f \end{cases}$$

Finally, $CoNet$ is a formulation of how we extract averaged feature vectors for X_i.

$$CoNet : \cup_{j=1}^n \gamma_j \longrightarrow \frac{\sum_{j=1}^n \sum_{k=1}^{|\gamma_j|} \frac{\overrightarrow{\Gamma}(\gamma_j^k)}{\sqrt[2]{\overrightarrow{\Gamma}(\gamma_j^k) \bullet \overrightarrow{\Gamma}(\gamma_j^k)}}}{\sum_{j=1}^n |\gamma_j|}$$

We skipped stemming in the preprocessing stage for given γ_j^k due to low accuracy achieved in our experiments. Investigations showed stemming decreases word usage frequency in the data and it leads to poor $CoNet$ vectors.

II. Latent Semantic Analysis (LSA) [9] is a statistical approach to extract relations among words by meaning of their contexts of use in documents. LSA can be accomplished by applying a low-rank Singular Value Decomposition (SVD) on the N-grams/TF-IDF matrices to reduce the number of rows while preserving the similar structure among columns.

Table 1. Optimized parameters for TF-IDF and N-gram

Language	Model	ngram_range	max_features	norm	strip_accents	max_df	sublinear_tf	stop_words
					Parameters			
English	TFIDF	(2,4)	3000	l2	False	0.7	True	english
	N-gram	(1,3)	3000	-	False	0.7	-	-
Spanish	TFIDF	(3,4)	6000	l2	False	0.2	True	spanish
	N-gram	(1,3)	6000	-	False	0.2	-	spanish

- **N-grams**, it converts a collection of text documents to a matrix of token counts as a term frequency(TF) representation. N-gram is trained for transforming data into matrix of M_{ngram}.

$$M_{ngram} : X^i \to V_d^i$$

- **N-grams with TF-IDF weighting** [10], it works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Words that are common in documents tend to have higher TF-IDF numbers than others. TF-IDF matrix representation of documents presents fingerprinted features for documents. TF-IDF is trained for transforming data into matrix of M_{tfidf}.

$$M_{tfidf} : X^i \to V_d^i$$

LSA is dimension reduction which is able to capture and represent significant components of the lexis and passage meanings. Also, this has the effect of reducing noise in the data as well as reducing the sparseness of the matrix. From these perspectives, we applied SVD to N-grams and TF-IDF matrices for dimensional reduction with a component number of 200. *SVD* is a formulation of dimension reduction for our case. *SVD* is a transformer of M_{tfidf} and M_{ngram} to latent space.

$$SVD : V_d^i \to V_{200}^i$$

We used scikit-learn [2] python library for our experiments and training N-grams models for both languages. Experimental searches have been done for tuning N-grams and TF-IDF parameters using a 5 and 10 fold cross-validation. Table 1 shows a summary of the best achieved parameters for both languages.

3.3 Input Representation

According to our experiments, single representations mainly are not able to perform well after achieving specific accuracy due to their features overlaps and similarities. We will discuss it in more detail. Regarding the hypothesis of combining, weak learners can boost performance. We hypothesis that combining representations must

do the same in most of the cases. To overcome single representation issues and to keep representation combination simple, **LSACoNet** has been introduced as a concatenation of representations. The Δ is a transformer which is able to represent a combination of feature vectors for given user tweets in 700 dimensions.

$$\Delta : X^i \longrightarrow (CoNet(X^i), SVD(M_{tfidf}(X^i)), SVD(M_{ngram}(X^i)))$$

$$\Delta(X^i) \in V^i_{700}$$

3.4 Model Architecture

A fully connected feed-forward neural network [5] (namely FCNN) introduced to tackle fake/real news spreader detection challenge. Proposed FCNN contains an input layer with 1024 neurons, ReLU activation, dropout, and BatchNormalization. Next, FCNN follows 3 hidden layers, each holding 256, 128, and 64 neurons respectively with sigmoid activation and an output layer with 2 neurons, and BatchNormalization. At the input layer, BatchNormalization set to normalize the combined features from different representations. To reduce thinking of the network, dropout has been used with a probability of 40% at the input layer. To compile network spars categorical cross-entropy, loss function has been utilized. As an optimizer, Adam applied with a learning rate of 0.002. The process of experimenting with Deep Neural Networks has been done using Keras [3] a deep learning API written in Python.

4 Experiments and Results

This year, task organizers have provided a training corpus[2]. The corpus is composed of documents in English and Spanish, where each document contains 100 tweets for each author. The statistics of this corpus are presented in Table 2.

4.1 Baselines

In order to compare the proposed methods, we implemented 3 baselines as described in bellow, and Table 3 (in group 0 for detailed experimental result) shows detailed evaluation results for them.

- RANDOM: a random prediction model predicts 1 if random value $\in [0, 0.5]$ else 0.
- TFIDFLSVM: TF-IDF representation contains all words without applying preprocessing and parameter tuning, and linear SVM as a classifier with $C = 1$.
- STATLSVM: includes statistical features like number of characters, URLs, Mentions, Hashtags, RTs, and Emojis with linear SVM as a classifier with $C = 1$.

---

[2] https://doi.org/10.5281/zenodo.3692319

Table 2. Dataset Statistics: Due to privacy reasons URLs, Mentions, and Hashtags are replaced with their specific tags starts/ends with #

| Dataset | Language | Class | User No. | Average Per User | | | | | | | Test |
				Tweets No.	Chars	Emojis	#URL#	RTs	#USER#	#HASHTAG#	
Train	English	1	150	100	8864	4	114.5	8	15.5	32	100
		0	150	100	8902	12	110.5	16	38.5	46	100
			300	100	8882	8	112	12	27	39	200
Train	Spanish	1	150	100	9113	12	93	15	40	11	100
		0	150	100	11136	28.5	73	29	71	39	100
			300	100	10124	20	83	22	55.5	25	200

4.2 Experimental Results

We conducted a few experiments with different classifiers (Multi-layer Perceptron, Linear/RBF SVM, Logistic Regression, K Nearest Neighbors, Naive Bayes, Ridge classifier - a classifier using ridge regression, Stacking Ensemble), and different representations(N-gram, TF-IDF, LSA, ConceptNet Numberbatch). The differences between experiments are mainly focused on 5/10-fold cross-validation mean accuracy and confidence interval(CI). Most of the models in experiments were suffering from a hight confidence interval. We essentially concentrated on reducing the overfitting impact by reviewing confidence intervals, while boosting model performance on validations using 5/10-fold cross-validation scheme.

Experiment 1: TF-IDF Modeling In Experiment 1 we used TF-IDF representation using word usage factor while making a vocabulary for representation. With word usage factor we were able to use the author's fingerprinted words as a representation with ignoring less and most used words with setting lower/upper bound threshold to each term frequency. We used a lower/upper bound term frequency thresholds for both languages. The lower/upper bound term frequency threshold includes 2/2000 for English and 3/4000 for Spanish. In final, terms fall in the range of $[L_{tf}, U_{tf}]$ considered in making TF-IDF vocabulary. Attained results for this experiment is recorded in Table 3 (in the section for detailed experimental results using cross-validation) group 1. We achieved CI close to 0.05 by applying a linear SVM classifier. The ridge classifier also achieved average accuracy result close to linear SVM, however, this model suffers from a high CI.

Experiment 2: Character N-gram Modeling In Experiment 2 similar to the previous analysis, we have run an investigation with character n-gram representation to explore for better features by keeping only the author's most valuable words. We used a character 3-grams scheme using word usage factor while making vocabulary for representation. Less valuable terms were ignored from the vocabulary by setting a

lower bound term frequency threshold of 5 for both languages. In final, terms fall in the range of $[L_{tf}, \infty)$ considered in making representation vocabulary. Accomplished results for this experiment were recorded in Table 3 group 2. Presented results are not very promising due to high CI and low accuracy regarding previous experiment models. Most importantly averaged results and CIs are close to baseline models except 2 cases and they are mostly suffering from high CI. More investigations revealed that for Spanish, logistic regression, and ridge classifiers are running well, however, for English, they are performing very low regarding baseline and group-1 models. According to the results, character n-gram representation fails in capturing fake/real news spreaders.

Experiment 3: Punctuation/Character N-gram Modeling In Experiment 3 we considered another study with character 5-grams with considering only marks. We replaced letters in tweets with *****. Next, we used the experiment 2 details for modeling logistic regression and linear SVM. Recorded results in Table 3 group 3 for both classifiers confirms that extracting character n-gram features could be hard for models to capture fake/real news spreaders due to poor features.

Experiment 4: Ensemble Learning In Experiment 4, we investigated combining weak learners by applying a stacking ensemble approach with a majority voting scheme. TF-IDF representation using linear SVM, k nearest neighbors, and ridge classifiers were considered for English, and at Spanish, only third learner changed to character 3-gram representation with logistic regression classifier. We achieved accuracies of **0.768/0.764** for averaged 5/10-fold cross-validation respectively. It outperforms current models, however, it suffers from high CI for English.

Experiment 5: Concept Modeling In Experiment 5, we examined linear/rbf SVM and logistic regression classifiers with ConceptNet Numberbatch word embedding. Obtained results are reported in Table 3, group 5. Results showed Numberbatch is mostly likely to perform similar to TF-IDF representation.

Experiment 6: Concatenation of Features In Experiment 6, we analyze LSACoNet representations with linear SVM. For analysis, we made a baseline without any specific parameter setting and using maximum feature dimensions. Interestingly we achieved a low CI for both languages with this baseline. It showed how combination of features are capable. Next, LSACoNet representations were evaluated based on the parameter setting mentioned in Table 1. Obtained results showed feature combination is a very powerful technique for boosting performance. We reached accuracies of **0.785/0.765** for 5/10-fold cross-validation and lowest possible CI. Detailed results have been recorded in Table 3 in group 6.

Experiment 7: FCNN In Experiment 7, we made a different analysis using LSACoNet representation and CoNet representation. To make conclusions about if

FCNN is able to perform better than the models described in previous experiments, CoNet representation is considered as a baseline. We used 5 different test split sizes for this experiment to evaluate LSACoNetFCNN and CoNetFCNN models. Obtained result from this experiment were recorded in Table 1 (in detailed experimental results with FCNN). We gained average accuracy of 0.79 for LSACoNetFCNN. Both results in experiment 6, and 7 are very promising, and comparing accuracies and CIs of these 2 experiments are not an interesting job to do because of differences in evaluations. Both LSACoNetLSVM and LSACoNetFCNN models are very promising and since for final evaluation, we didn't have any test set to compare these 2 models we simply relied on LSACoNetFCNN as a final model.

4.3   Final Evaluation

Following the previous results, for the final evaluation at TIRA platform [14], we applied LSACoNet method with FCNN for the classification of real/fake news spreaders. The obtained accuracy results for the final evaluation were as follows: in Spanish, 0.745; in English, 0.725; and 0.735 for both tasks. The official results are shown in Table 3 (in detailed results of submissions) for early birds and final evaluation. We gained a better result with LSACoNet and FCNN for English at the final evaluation. However, for Spanish TF-IDF representation with linear SVM performed well with an accuracy of 0.765 at early birds evaluation. In the final evaluation metrics, the best scores of the submissions between the early birds and final submissions of each participant and each language have been considered. This means that in our case we achieved the best score for Spanish in early bird and the best score for English in the final submission so, overall achieved accuracy is **0.745**.

5   Conclusions

In this paper, we proposed a model for Profiling Fake News Spreader on the Twitter task in PAN 2020. We presented a feature combination model namely LSACoNet to use a different representation of the documents to incorporate with FCNN on detecting fake/real news spreaders on Twitter. In the final, we achieved average accuracy of **0.745**. Regarding our manual evaluation, our approach is very capable of distinguishing fake/real news spreaders. In future works, we most likely to try to add feature weighting for representations and use different deep neural network models like RNN and cleverly emotionalized word or character n-gram features to enrich current features to boost the performance of currently existed representation.

References

1. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
2. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn

Table 3. Detailed experimental results using Cross-Validation (5CV/10CV = 5/10-fold Cross Validation), FCNN with test size of 0.1, 0.2, 0.3, 0.4, 0.5 percent from training-set, and Detailed results of submissions. Also, models tagged with $<en>$ and $<es>$ are chosen models for early birds submission. Moreover, tag $<BS>$ was used to show baseline models.

Detailed Experimental Results using Cross-Validation

Group	Model	English 5-CV Mean	CI	English 10CV Mean	CI	Spanish 5CV Mean	CI	Spanish 10CV Mean	CI	Average 5CV	10CV
0	RANDOM$^{<BS>}$	0.560	0.090	0.493	0.159	0.513	0.082	0.539	0.183	0.536	0.516
	TFIDFLSVM$^{<BS>}$	0.666	0.109	0.633	0.186	0.760	0.108	0.766	0.149	0.710	0.699
	STATLSVM$^{<BS>}$	0.613	0.108	0.613	0.169	0.723	0.050	0.720	0.120	0.668	0.666
1	LSVM$^{<es>}$	0.710	**0.068**	0.693	0.122	0.820	**0.048**	0.826	0.077	**0.765**	**0.758**
	LSALSVM	0.700	0.105	0.679	0.133	0.823	0.054	0.816	0.095	0.761	0.745
	KNN	0.683	0.096	0.700	0.073	0.756	0.074	0.740	0.106	0.719	0.720
	LSAKNN	0.613	0.108	0.653	0.169	0.720	0.032	0.720	0.112	0.666	0.686
	MLP	0.696	0.133	0.679	0.155	0.806	0.054	0.799	0.119	0.751	0.739
	NB	0.706	0.088	0.693	0.135	0.736	0.104	0.743	0.115	0.721	0.718
	RIDGE	0.713	0.104	0.700	0.115	0.820	0.053	0.840	0.083	**0.766**	**0.770**
	LSARIDGE	0.696	0.082	0.676	0.094	0.806	0.061	0.826	0.071	0.751	0.751
2	LR	0.650	0.063	0.663	0.194	**0.823**	**0.033**	0.790	0.099	0.736	0.726
	LSALR	0.603	0.099	0.673	0.145	0.776	0.108	0.766	0.103	0.689	0.719
	KNN	0.550	0.124	0.600	0.163	0.743	0.140	0.753	0.149	0.646	0.676
	LSAKNN	0.543	0.071	0.586	0.195	0.750	0.119	0.753	0.176	0.646	0.669
	RIDGE	0.656	0.071	0.696	0.182	0.820	0.038	0.806	0.139	0.738	0.751
	LSARIDGE	0.636	0.176	0.646	0.221	0.776	0.045	0.780	0.133	0.706	0.713
3	LR	0.610	0.093	0.636	0.205	0.753	0.116	0.750	0.104	0.681	0.693
	LSVM	0.610	0.112	0.636	0.164	0.723	0.143	0.723	0.115	0.666	0.679
4	ensemble	0.713	**0.092**	0.696	0.117	0.823	**0.049**	0.833	**0.073**	0.768	0.764
5	LSVM$^{<en>}$	0.683	0.101	0.669	0.128	0.726	0.045	0.730	0.109	0.704	0.699
	RBFSVM	0.706	0.074	0.706	0.118	0.779	0.082	0.773	0.162	0.742	0.739
	LR	0.713	0.106	0.713	0.149	0.733	0.059	0.736	0.113	0.723	0.724
6	LSVM$^{<BS>}$	0.673	**0.029**	0.690	**0.058**	0.770	0.025	0.810	**0.060**	0.721	0.750
	LSVM	**0.750**	0.037	**0.730**	0.068	0.820	**0.022**	0.800	0.061	**0.785**	**0.765**

Detailed Experimental Results with FCNN

	Model	Lang	0.1	0.2	0.3	0.4	0.5	Mean	CI	Description
7	CoNetFCNN$^{<BS>}$	en	0.70	0.63	0.71	0.66	0.65	0.67	0.060	FCNN baseline
	CoNetFCNN$^{<BS>}$	es	0.83	0.80	0.74	0.74	0.79	0.78	0.070	FCNN baseline
	LSACoNetFCNN	en	0.76	0.81	0.80	0.79	0.75	**0.78**	**0.044**	Final Model
	LSACoNetFCNN	es	0.80	0.80	0.80	0.81	0.82	**0.80**	**0.018**	Final Model

Detailed Results of Submissions

	Submission Type	English Model	Accuracy	Spanish Model	Accuracy	Average
8	Early Birds Submission	CoNetLSVM	0.685	TFIDFLSVM	**0.765**	0.725
	Final Submission	LSACoNetFCNN	**0.725**	LSACoNetFCNN	0.745	**0.735**

project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
3. Chollet, F., et al.: Keras. https://keras.io (2015)
4. Ghanem, B., Rosso, P., Rangel, F.: Stance detection in fake news a combined feature representation. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). pp. 66–71 (2018)
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
6. Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C.M., Gurevych, I.: A retrospective analysis of the fake news challenge stance-detection task. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 1859–1874. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1158
7. Hanselowski, A., S., A.P.V., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C.M., Gurevych, I.: A retrospective analysis of the fake news challenge stance detection task. CoRR abs/1806.05180 (2018), http://arxiv.org/abs/1806.05180
8. Karadzhov, G., Gencheva, P., Nakov, P., Koychev, I.: We built a fake news & click-bait filter: What happened next will blow your mind! CoRR abs/1803.03786 (2018), http://arxiv.org/abs/1803.03786
9. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse processes 25(2-3), 259–284 (1998)
10. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, USA (2008)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013), http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf
12. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
13. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. arXiv preprint arXiv:1708.07104 (2017)
14. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World. The Information Retrieval Series, Springer (Sep 2019)
15. Rahgouy, M., Giglou, H., Rahgooy, T., Sheykhlan, M., Mohammadzadeh, E.: Cross-domain Authorship Attribution: Author Identification using a Multi-Aspect Ensemble Approach. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019), http://ceur-ws.org/Vol-2380/
16. Rangel, F., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings (Sep 2020), CEUR-WS.org
17. Riedel, B., Augenstein, I., Spithourakis, G.P., Riedel, S.: A simple but tough-to-beat baseline for the fake news challenge stance detection task. CoRR abs/1707.03264 (2017), http://arxiv.org/abs/1707.03264
18. Shu, K., Wang, S., Liu, H.: Understanding user profiles on social media for fake news detection. In: 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). pp. 430–435 (2018)

19. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An open multilingual graph of general knowledge. pp. 4444–4451 (2017),
http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972
20. Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Chair), N.C.C., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association (ELRA), Istanbul, Turkey (may 2012)
21. Volkova, S., Shaffer, K., Jang, J.Y., Hodas, N.: Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 647–653 (2017)