Noname manuscript No.
(will be inserted by the editor)

Twitter alloy steel disambiguation and user relevance via one-class and two-class news titles classifiers

Paola Zola · Paulo Cortez · Eugenio Brentari

Received: date / Accepted: date

Abstract This paper addresses the nontrivial task of Twitter financial disambiguation (TFD), which is relevant to filter financial domain tweets (e.g., alloy steel or coffee prices) when no unique identifiers (e.g., cashtags) are adopted. To automate TFD, we propose a transfer learning approach that uses freely labeled news titles to train diverse one-class and two-class classification methods. These include different text handling transforms, adaptations of statistical measures and modern machine learning methods, including support vector machines (SVM), deep autoencoders and multilayer perceptrons. As a case study, we analyzed the domain of alloy steel prices, collecting a recent Twitter dataset. Overall, the best results were achieved by a two-class SVM fed with TFD statistical measures and topic model features, obtaining an 80% and 71% discrimination level when tested with 11,081 and 3,000 manually labeled tweets. The best one-class performance (78% and 69% for the same test tweets) was obtained by a term frequency-inverse document frequency classifier (TF-IDFC). These models were further used to generate a Financial User Relevance rank (FUR) score, aiming to filter relevant users. The SVM and TF-IDFC FUR models obtained a predictive user discrimination level of 80% and 75% when tested with a manually labeled test sample of 418 users. These results confirm the proposed joint TFD-FUR approach as a valuable tool for the selection of Twitter texts and users for financial social media analytics (e.g., sentiment analysis, detection of influential users).

Keywords Text classification · User relevance · Machine learning · Social Media analytics

Paola Zola
IIT-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy.
E-mail: paola.zola@iit.cnr.it

Paulo Cortez
ALGORITMI Centre, Department of Information Systems, University of Minho,
4804-533 Guimarães, Portugal.

Eugenio Brentari
Department of Economy and Management, University of Brescia, Brescia, Italy.

1 Introduction

More than 300 million people use Twitter every month, resulting in 500 million tweets sent each day[1]. Thus, Twitter is a powerful big data source of freely opinionated texts for social media analytics, with a wide range of applications, including political sentiment analysis [1] or inferring the user country of interest [2].

In particular, there has been a recent research trend of using social media sentiment analysis for financial decision support systems [3,4]. Regarding Twitter, the most common approach to retrieve texts is based on a keywords match by using the application programming interface (API). Using such API it is easy to extract tweets about stock markets, since specific company cashtags are commonly used (e.g., the cashtag $AAPL univocally identifies the Apple technology stock prices) [3]. As shown in Table 1, several research studies used these unique cashtag identifiers to analyze the sentiment of tweets related to company stocks or indexes (e.g., [5,3]). However, research addressing the sentiment of alloy or commodity prices is scarce and mainly considers texts from authoritative sources, such as Thomson Reuters [6,7]. In fact, Twitter sentiment analysis in this domain is not as simple as for financial stocks, since alloy and commodity texts do not typically have a unique ticker. Thus, a generic keywords search needs to be used (e.g., *silver prices*). Yet, this often results in misleading tweets. This problem was recently pointed out by [4], which detected a large amount of noisy tweets when using generic keywords for filtering stock index futures and thus needed to adopt a manually curated list of known financial experts to filter the data.

As a demonstration example, we extracted three sets of tweets (each with 100 texts) by using the keywords *cocoa*, *silver price* and *steel price*. After a manual inspection of the tweets, we found that only 13%, 43% and 47% of the tweets were related to cocoa, silver and steel in sense of financial materials. When using the keyword *steel price*, four of the extracted tweets were:

1. "us stainless steel sheet prices moved up to start april as mills lowered base price discounts and demand increased";
2. "galvanized steel sheet roofing corrugated iron prices";
3. "sale stainless steel commercial kitchen list price";
4. "low prices on our top selling cylinder blanks in brass steel follow link below".

All four tweets are related with steel products but only the first two refer to steel industrial production. In effect, the last two are relevant for retail consumers and thus should be discarded when executing alloy steel price analytics. Word sense disambiguation (WSD) methods, which disambiguate words based on lexicons (e.g., *commercial bank* versus *river bank*), do not distinguish well these tweets. For instance, when we apply the known Lesk WSD [17], the resulting synsets classify all four tweets as not related to alloy steel.

Within our knowledge, no studies have performed Twitter sentiment analysis of alloy or commodity prices (Table 1), which is probably due to the difficulty of retrieving the relevant texts. To solve this issue, this paper introduces the concept of Twitter financial disambiguation (TFD), which can be seen as a form of text classification specifically built for filtering financial tweets when the search keyword has an unique meaning but that can be related with different contexts (e.g., *steel*

[1] `https://blog.hootsuite.com/twitter-statistics/`

Table 1 Financial domain sentiment analysis studies.

Study	**Target**[a]	**Markets**[b]	textbfTextual[c] {**Data**	**Period**
Bollen et al.[8]	SI	DJIA	TW	2008-2008
Lechthaler et al.[6]	CF	CO	TR	2003-2010
Feuerriegel et al.[9]	CF	G, CO	TR	2003-2012
Rao et al.[10]	SI,CF,F,VIX	DJIA, NAS, CO, G, EUR/USD	TW,SVI	2010-2011
Prollochs et al.[11]	SI	TRD	RA	2004-2011
Nguyen et al.[12]	S	18S	YFMB	2012-2013
Pagolu et al. [5]	S	MS	TW	2015-2016
Li et al.[7]	CF	CO	TR	2008-2014
Oliveira et al.[3]	S,P	SP, RSL, RMRF, DJIA, NAS, HML, , SMB, VIX, PInd, PSize	TW	2012-2015
Daniel et al.[13]	S	DJIA	TW	2013-2015
Maslyuk et al.[14]	CF	CO, NG, PR, GAS, HO	TR	2003-2014
Huang et al.[15]	S,B,CF,F,H	SP, HPI, 3-YGB, USD, TRC	TR	1998-2016
Mudinas et al.[16]	SI,S,F	DJIA, AAPL, GOOGL, HP, JPM, EUR/USD, GBP/USD	FT,Re,TW	2011-2014
Gross et al.[4]	SIF	Europe, USA, Asia and Australia	TW	2010-2018

[a] B: Bond, CF: Commodity Futures, F: Forex, H: Housing prices, P: Portfolio, S: Stocks, SI: Stock Index, SIF: Stock Index Futures, VIX: Volatility Index.

[b] 18S: 18 different Stocks quoted on DIJA, 3-YGB: 3 Years Government Bond, AAPL: Apple, CO: Crude Oil, DJIA: Dow Jones Industrial Average, EUR/USD: forex euro US dollar, G: gold, GAS: gasoline, GOOGL: google, GBP/USD: forex Great British Pound and US dollar, HO: heating oil, HML: high minus low, HP: Hewlett-Packard, HPI: housing price index, JPM: J.P. Morgan Chase & Co, MOM: momentum factor, MS: Microsoft, NG: natural gas, NAS: Nasdaq, PInd: 10 Industry Portfolio, PR: propane, PSize: Portfolio formed on size, SMB: small minus big, SP: S&P500, RMRF: excess return on the market, RSL: Russell 2000, TRC: Thomson Reuters commodity prices, USD: US dollar currency, VIX: volatility Index.

[c] FT: Financial Times, TR: Thomson Reuters, TW: Twitter, RA: Regulatory Announces, Re: Reddit, SVI: Search Volume Index from Google, YFMB: Yahoo Finance Message Board.

sheet versus *steel kitchen*). As a case study, we consider alloy steel prices, which is a financially relevant domain. Steel is the fourth most commonly used metal in the world and it is highly important to the global economy, since trends in production are an indicator of the health of a country's economy[2]. In the United States, the steel industries employee around $142,000$ people and about 6.5 million Americans are employed by steel-consuming companies[3]. Traditional attempts to study alloy steel prices employ classical time series analysis [18] or analyze the extraction patterns from iron and coal mines [19,20], as well as energy, transportation and products storage costs [21].

To address the TFD task, we propose an automatic transfer learning approach [22], in which freely available labeled news titles are used to train diverse text classifiers. Two main transfer learning strategies are explored, based on having access to a training set of news titles with only positive financial texts (one-class classification) or with positive and negative examples (two-class case). For the former strategy, we adapt different distance measures (cosine and dynamic time warping), autoencoders (simple and deep learning), a term frequency-inverse document frequency classification (TF-IFC) measure and a one-class support vector machine (OC-SVM). For the latter strategy, we adapt several distance and statistical measures (e.g., cosine, information gain, TF-IDFC) and also explore three supervised machine learning (ML) algorithms: random forest (RF), support vector machine

[2] `https://www.focus-economics.com/blog/steel-facts-commodity-explainer`

[3] `https://money.cnn.com/2018/03/07/news/companies/trump-tariffs-steel-jobs/index.html`

(SVM) and deep multilayer perceptron (MLP). All TFD methods generate a relevance score for each tweet. We aggregate these scores, aiming to create a financial user relevance rank (FUR) score, which indicates the degree of relevance of a user, thus being useful for filtering users (e.g., Twitter users that are interesting to follow). As explained in Table 3, most research studies measure user influence or expertise by adopting specific user data (e.g., metadata, historical tweets) or social network graph analysis. The novelty of the FUR score is that it only considers the texts retrieved by the keywords query, thus it does not require an access, storage and analysis of user metadata, historical tweets or social network interaction data. The main contributions of our joint TFD-FUR approach are:

1. we address the TFD task, focusing on the case study of alloy steel prices;
2. we use freely and easily available news titles to compute the TFD models, thus making use of a transfer learning approach that avoids a costly human labeling;
3. we compare several TFD one-class and two-class learning approaches that are based on novel adaptations of statistical measures and modern ML algorithms;
4. we propose a new FUR score that only considers the texts returned by a keywords Twitter query;
5. we collect and analyze a recent alloy steel Twitter dataset that is publicly made available for further TFD researches.

The paper is structured as follows. Section 2 details the related work about text classification and user relevance. Then, Section 3 describes the proposed approach, which includes TFD and FUR methods. Next, Section 4 reports the data used (Section 4.1), experiments performed and the obtained results (Sections 4.2 and 4.3). Finally, Section 5 discusses the main conclusions.

2 Related work

2.1 Twitter financial disambiguation

The TFD concept is associated with the research topics of text similarity (TXS), WSD and topic modeling (TM), all related to text classification. Table 2 summarizes the most relevant studies covering these topics, assuming a chronological order and a particular focus on short texts, as provided by microblogs. The Table contains the following columns: **Aim** – the main research topic (TXS, WSD, TM or TFD); **Learning** – use of unsupervised or supervised learning (with labeled data); **Text size** – use of long or short (microblog) texts; **Training source** – data used to tune or train the method (if any and when different from target source); **Token handing** – preprocessing method used to handle the texts; **Model** – model adopted for the research topic; **Target source** – data where the model was validated; **Metrics** – model performance metrics; and **Validation** – type of validation method (e.g., $k-$fold cross validation, rolling window).

Measuring the similarity between two texts (TXS) is a nontrivial task, especially if the texts have different sizes and include slang or abbreviations, often used in short microblog messages. TXS is often achieved by computing a text similarity measure. The most common measures are [36]: Euclidian distance, Jaccard similarity and Cosine Distance. Yet, these traditional measures require vectors with the same length. To solve this issue, [23] used dynamic time warping (DTW) for TXS.

Table 2 Summary of the related work for Financial Twitter Disambiguation (TFD).

Study	Aim[a]	Learning[b]	Text size[c]	Training source[d]	Token handling[e]	Model[f]	Target source[g]	Metrics[h]	Validation[i]
Banerjee et al. [17]	WSD	U	-	WN	STR	Lesk	SensEval-2	ACC	-
Liu et al. [23]	TXS	U	S	WN	STR	DTW	-	COR	-
Yan et al. [24]	TM	U	S	-	STR	BTM	Tweets2011, Q&A, 20NewsGroup	ACC,Purity, NMI,ARI	5-CV
Iosif et al. [25]	TXS	U	L	YS,G	STR, BOW, TFIDF	PCTXS	Charls Miller, MeSH	COR	-
Kenter et al.[26]	TXS	S^2	S	AWE	W2V, WE	SVM	MSC	ACC	-
Song et al. [27]	TXS	U	S	-	W2V	DESA	[28], ACE2005, [29]	ACC,F1,COR	5-CV
Zhang et al. [30]	WSD	U	L,S	Wiki	STR	LMSK	AQUAINT, Blog06	MAP	-
Amiri et al. [31]	TXS	U,$S^{>2}$	-	-	WE	CS AE	SCWS, Q&A	MAP,MRR	HO
Neculoiu et al. [32]	TXS	$S^{>2}$	S	-	WE	SiRNN	Job titles	ACC	-
Lim et al. [33]	TM	U	S	-	STR	ClusTop	Twitter	TC,PMI, P,R,F1	4-CV
Chaplot et al. [34]	WSD	U	-	WN	STR	WSDTM	SemEval	F1	-
Li et al. [35]	TM	U	L,S	-	STR	EW	8 datasets	ACC, NMI	5-CV
This paper	**TFD**	**$S^{1,2}$**	**S**	**NT**	**W2V, STR, TF-IDF**	**TF-IDFC, CD,DTW, SiAE, IG,PMI, RF,SVM, MLP**	**Twitter**	**AUC**	**RW**

[a] TFD: Financial Twitter Disambiguation, TM: Topic Modeling, TXS: Text Similarity, WSD: Word Sense Disambiguation.
[b] S: Supervised (1 – one-class texts; 2 – two-class texts; > 2 – more than 2 classes), U: Unsupervised.
[c] L: Long text, S: Short text.
[d] AWE: Augmented Word Embedding, NT: News Titles, YS: Yahoo search, G: Google, Wiki: Wikipedia, WN: WordNet.
[e] BOW: Bag of Words, STR: String, TF-IDF: Term-Frequency Inverse-Document-Frequency matrix, WE: Word Embedding, W2V: Word2Vec.
[f] BTM: Biterm Topic Model, CD: Cosine Distance, CS AE: Context Sensitive Autoencoder, DESA: Dense Explicit Semantic Analysis, DTW: Dynamic Time Warping, EW: Entropy Weighting, LMSK: Language Model and Structural Knowledge, MLP: Multilayer Perceptron, PCTXS: Page Count and Text Based Similarity, SiAE: Siamese Autoencoder, SiRNN: Siamese RNN, SVM: Support Vector Machine, TF-IDFC: TF-IDF Classifier, WSDTM: Word Sense Disambiguation Topic Modelling.
[g] MeSH: Medical Subject Headings, MSC: Microsoft Paraphrase Corpus, Q&A: Question and Answering corpus, SCWS: Word similarity dataset.
[h] ACC: Accuracy, ARI: Adjusted Rand Index, AUC: Area Under the receiver operating characteristic Curve, COR: correlation, F1: F1-score, MAP: Mean Average Precision, MRR: Mean Reciprocal Rank, PMI: Pointwise Mutual Information, NMI: Normalized Mutual Information, P: Precision, R: Recall, TC: Topic Coherence,
[i] HO: Holdout train and test split, k-CV: k-fold Cross Validation, RW: Rolling Window.

Other approaches used augmented Web documents [25]. The use of augmented texts is also often adopted for WSD tasks (e.g., WordNet lexical database) [23, 34]. Moreover, the WSD works from Table 2 combine features extracted using a TM algorithm. The latent Dirichlet allocation (LDA) [37] is a popular algorithm for TM. More recently, the biterm topic model (BTM) method was proposed, aiming to achieve a better TM for short texts [24]. In Table 2, the initial studies were mainly based on string comparisons, with the original words. Recent TXS works use a word embedding (e.g., Word2Vec) to get a numerical representation of the texts [27,32]. Only the most recent studies employ deep learning models, such as recurrent neural networks [38] and autoencoders [31].

The approach proposed in this paper appears at the last row of Table 2. Our approach differs from the ones in the Table 2 since it is specifically built for financial tweets already filtered by specific keywords. Only one other study adopted Twitter [33], performing a topic clustering based on networks of words that automatically define the number of topics, using a series of tweet features (e.g., hastags, mentions and nouns). Moreover, most supervised learning studies used binary labels, while we approach two training setups: one-class (unary), in which only positive financial texts are available; and two-class (binary), which assumes an access to both positive (financial) and negative (non financial) messages. Since Twitter texts

are unlabeled, and in order to avoid a laborious manual effort, we use public and freely available news titles to set the positive and negative messages, thus making use of a transfer learning [22,39]. As for the TFD models, we adjust and compare several data preprocessing, statistical measures and ML algorithms, including recent Word2Vec encoding and deep learning methods (e.g., siamese autoencoder, deep multilayer perceptron). The models are evaluated using a robust and realistic rolling window procedure [40,3].

2.2 Social media user relevance

In general, there are two main ways to measure what is an influential or relevant social media user: based on user social network features or user data (e.g., metadata, historical texts). Table 3 surveys these influential user research approaches, with a particular focus on studies that analyze one specific user relevance topic, as our case study. Table 3 includes the columns: **Model** – proposed model to measure user relevance; **User network** – based on the usage of social network attributes (e.g., followers); **User history** – based on the usage of user metadata or historical messages; **Target source**, **Metrics** and **Validation** – similar meaning of Table 2.

Table 3 Summary of the related work for Financial User Relevance (FUR).

Study	Model[a]	User network	User history	Target source	Metrics[b]	Validation[c]
Yamaguchi et al. [41]	TuRank	X	-	Twitter	AA	-
Castillo et al. [42]	Fea,SVM, DT,BN	-	X	Twitter	MAE,P,R, ACC,F1	3-CV
Pal et al. [43]	Fea,GMM	X	-	Twitter	P,R,COR	-
Gayo et al. [44]	PD	X	-	Twitter	Min,Med Mean	-
Ito et al. [45]	LDA,Fea, RF	-	X	Twitter	AUC	10-CV
Cortez et al. [46]	Fea	X	-	StockTwits	COR, PQU	RW
Eliacik et al. [47]	Fea,PgR	X	-	Twitter	COR	10-CV
This paper	**TFD**	**-**	**-**	**Twitter**	**AUC**	**RW**

[a] BN: Bayesian Network, DT: Decision Trees, Fea: Feature Analysis, GMM: Gaussian Mixture Model, LDA: Latent Dirichlet Allocation, PD: Paradoxical Discounted, RF: Random Forest, SVM: Support Vector Machine, TFD: Financial Disambiguation based.

[b] AA: Average Adequacy, ACC: Accuracy, AUC: Area Under the receiver operating characteristic Curve, COR: Correlation, F1: F1-score, Min: Minimum, Med: Median, P: Precision, PQU: Percentage of Quality Users, R: Recall.

[c] k-CV: k fold Cross Validation, RW: Rolling Window.

Most studies of Table 3 focus on Twitter. Also, the state-of-the-art works assume two major sources of data: social networks (e.g., graphs of user interactions) and/or user history (e.g., metadata, user past tweets). The former source is often modeled by using graph network analysis, computing measures such as indegree or Page Rank [43,46]. The latter involves specific user metadata attributes, such as age [42], or access to user past tweets [45]. The novelty of our FUR approach

(shown in the last row of Table 3) is that it works directly over the messages retrieved from a keywords query, with no need to access social network or user history data.

3 Methods

3.1 Problem Statement

Recent trends in financial commodities price predictions have overcome the traditional usage of quantitative features introducing textual information. However, as explained in Section 1, textual information, especially from social media involve text processing issues that need to be carefully addressed.

In this paper we focus on the following TFD task. Let Q denote a set of tweets that resulted from applying a K keywords query to a Twitter API service. The goal of K, manually defined by an financial analyst, is to retrieve all Twitter messages related with a financial domain market F. However, Q often contains irrelevant texts (as shown by the *cocoa*, *silver* and *steel* price query examples of Section 1). Thus, the TFD task consists in an automatic filtering of the Q messages, leading to a subset Q_F that contains a higher number of texts that are relevant to the F domain, thus useful for social media financial analyses (e.g., sentiment analysis). A learning model for TFD can be fit by using a training sample with: only Positive texts (P) related with F (one-class classification); or Positive (P) and Negative example texts ($P \cup N$, two-class classification). Once a TFD learning model is fit, it can be used to compute a relevance score S_t for a particular tweet t, where $t \in Q$. These individual scores can be aggregated, allowing to compute a FUR score for a particular user u, which can be used as a measure of relevance of u within the F domain. The advantage of FUR is that it only requires the keywords query texts (Q) to filter the F domain experts, thus avoiding the need to retrieve and process user metadata and other historical tweets.

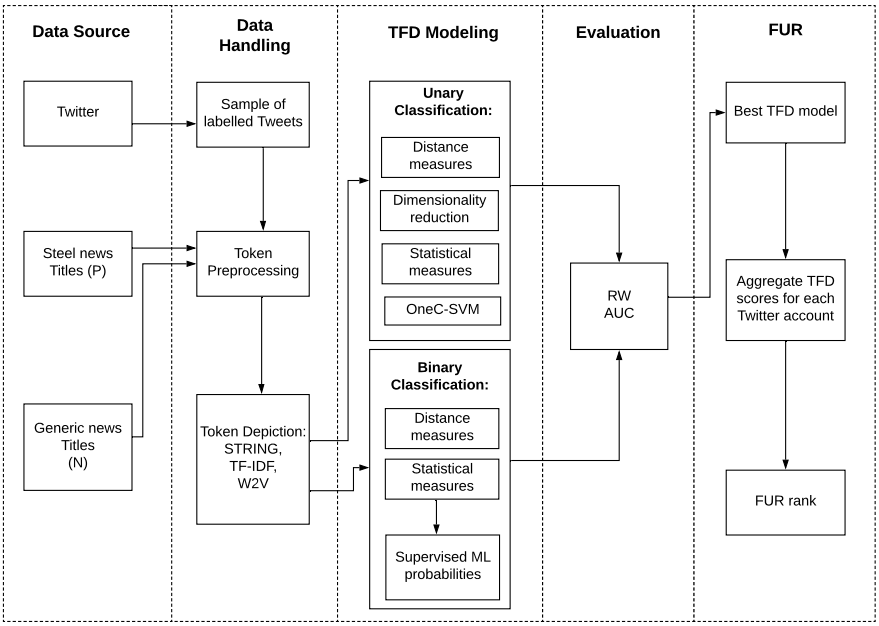
3.2 Proposed Approach

The proposed approach for TFD and FUR is depicted in Figure 1 and it includes five main steps: data source, data handling, TFD modeling, evaluation and FUR.

First, a Twitter keywords search is executed, resulting in a set of tweets that should be related with a F financial topic but that also include other irrelevant texts. As a case study, this paper addresses the F = *alloy steel prices* domain. For TFD, this paper adopts a supervised learning, under two main approaches: one-class (unary) and two-class (binary) classification. In order to get labeled data for train the models, we use easy to collect and freely available news titles (as detailed in Section 4.1). Given the samples of P and N, the TFD models use a transfer learning [22,39], where the models are adjusted to one training source (news titles) and tested on a different source (Twitter).

In the second step, the collected tweets and news titles are preprocessed. All texts (news titles and tweets) are transformed into a lowercase representation removing punctuation and stop words (e.g., "the", "and"). The resulting tokens

Fig. 1 Schematic of the research approach for TFD and FUR.



might be used directly (as string) or further processed into a numeric representation, via a term frequency-inverse document frequency (TF-IDF) matrix or Word2Vec (W2V) transform.

TF-IDF is a common transform for texts [48] that is computed as:

$$\begin{aligned} tf_{i,j} &= \frac{n_{i,j}}{n_{d_j}} \\ idf_i &= \log \frac{n_D}{n_{d:i \in d}} \\ tf\text{-}idf_{i,j} &= tf_{i,j} \times idf_i \end{aligned} \tag{1}$$

where $n_{i,j}$ is the number of occurrences of token i in document d_j, n_{d_j} is the number of tokens in document d_j, n_D is the number of documents in the collection and n_d is the number of documents in the collection that contain token i. W2V is a modern word encoding method that was proposed by [49]. W2V is based on a multilayer perceptron neural network with an input, projection and output layer. This work uses the unsupervised W2V algorithm with continuous bag-of-words (CBOW) model that is implemented at the `gensim` module in Python. The algorithm includes only one hyperparameter, the embedding size E (vector size for each token). To fix the hyperparameter, the embedding size is ranged within the values $E \in \{1, 8, 16, 32\}$.

All tested supervised ML methods (RF, MLP, SVM) and autoencoders require a fixed input size but the analyzed texts include a variable number of tokens. To handle this issue, when using direct text token inputs (TF-IDF or W2V), the truncation technique employed in [50,39] is adopted, which considers only the

first M tokens, as they appear in the texts. If the texts have less than M tokens then we use padding, which consists in adding null values (e.g., 0) [39]. Thus, supervised ML algorithms and autoencoders assume M inputs when using the TF-IDF transform and $E \times M$ inputs when the W2V encoding is adopted.

The third step performs the TFD, under a one-class or two-class classification. One-class methods include: cosine distance (CD) and Dynamic Time Warping (DTW) distance measures; dimensionality reduction via autoencoders; a TF-IDF based statistical measure; and one-class SVM (OC-SVM), which is a popular ML algorithm for unary classification [51]. As for the two-class methods, they include: CD and DTW distance measures; a higher range of adapted statistical measures, namely TF-IDF based, information gain (IG) and pointwise mutual information (PMI); and binary supervised ML algorithms, namely RF, SVM and MLP.

The fourth step is detailed in Section 3.3. It involves the usage of a realistic rolling window (RW) evaluation, which includes several train and test model updates through time. The TFD method predictions are contrasted with a tweet labeled sample ground truth, allowing the computation of the area under the receiver operating characteristic curve (AUC).

Finally, in the fifth step, the best TFD model is selected and used to score all tweets. For each distinct Twitter user account, the scores are aggregated, resulting in the FUR rank (Section 3.2.3).

3.2.1 One-class methods for Twitter Financial Disambiguation (TFD)

The unary methods assume a training data composed by only positive texts (P). In this paper, these texts are represented by steel domain news titles (Section 4.1). The one-class models output a TFD relevance score (S_t), which is computed as presented in Table 4. The S_t can be interpreted as the degree of proximity of the tweet t to the training data. Thus, the higher is the S_t, the higher is the probability that the tweet t is related to the positive concept. For a binary classification, it is possible to label a text (or tweet) t as a positive class (value of 1) if the respective relevance score is $S_t > T_{\text{TFD}}$, where T_{TFD} is a decision threshold that can range through any value of the S_t function domain; otherwise t is considered as belonging to the negative class (value of 0).

In this paper, we adapt two distance measures: the classical CD and DTW. DTW is popular for time series analysis and it can handle texts with different sizes, without the need of padding, as required by the CD measure [52]. The relevance scores proposed in Table 4 allow to directly use CD and DTW as TFD one-class classifiers.

Since the analyzed texts have different dimension sizes, we also adopt a dimensionality reduction algorithm. Autoencoders (AE) are a type of generative neural network in which the output is the same as the input. In particular, we use the Siamese autoencoder (SiAE) [53]. The SiAE is trained using positive texts (P), using as inputs the TF-IDF or W2V encoded numerical values. It also includes a squeezed hidden layer, which allows to reduce the texts. After the SiAE structure is trained, it can be used to compress any new texts, including tweets. Two SiAE structures are explored (Table 5): a simpler one, with just one encoder and decoder layer with hidden size equal to 1 (Model number 0), and a Deep SiAE, with several hidden layers (10 distinct structures are tested, from Model number 1 to 10). The SiAE networks can directly perform a TFD unary classification by

Table 4 TFD relevance scores when using unary (P) or binary ($P \cup N$) texts.

TFD model	Token handling	Training	TFD relevance scores $(S_t)^a$
CD	TF-IDF,W2V	unary	$\sum_{u \in P} \frac{t \cdot u}{\|t\| \cdot \|u\|}$
		binary	$\frac{1}{n_P} \sum_{u \in P} \frac{t \cdot u}{\|t\| \cdot \|u\|} - \frac{1}{n_N} \sum_{v \in N} \frac{t \cdot v}{\|t\| \cdot \|v\|}$
DTW	TF-IDF,W2V	unary	$-\sum_{u \in P} DTW(t, u)$
		binary	$\frac{1}{n_N} \sum_{v \in N} DTW(t, v) - \frac{1}{n_P} \sum_{u \in P} DTW(t, u)$
SiAE	TF-IDF,W2V	unary	$-\sum_{u \in P} \|\boldsymbol{h}_t - \boldsymbol{h}_u\|$
TF-IDFC	TF-IDF	unary	$\sum_{i \in t} tf\text{-}idf_{i,t}$
		binary	$\sum_{i \in t} [(tf\text{-}idf1_{i,t}) - (tf\text{-}idf0_{i,t})]$
IG	string	binary	$\sum_{i \in t} IG(i)$
PMI	string	binary	$\sum_{i \in t} PMI(i, 1) - PMI(i, 0)$

[a] $tf\text{-}idf$ – TF-IDF computed using the positive ($tf\text{-}idf1$) or negative ($tf\text{-}idf0$) texts; n_P – number of positive financial texts; n_N – number of negative texts; DTW – DTW distance function; $\boldsymbol{h}_t$ autoencoder function for text t; $IG(i)$ – IG function for token i; PMI – PMI function computed for token i and positive (1) or negative (0) classes.

using the relevance score proposed in Table 4, where $\boldsymbol{h}_i$ denotes the autoencoder squeezed hidden layer function for text i.

Table 5 Different SiAE models compared.

SiAE number	Hidden layer size (h1)	Hidden layer size (h2)	Hidden layer size (h3)	Hidden layer size (h4)	Hidden layer size (h5)	Hidden layer size (h6)
0	-	-	-	-	-	1
1	6	5	4	3	2	1
2	10	7	5	4	3	1
3	25	20	15	10	5	1
4	50	40	30	20	10	1
5	150	50	25	10	5	1
6	5	2	-	-	-	1
7	10	5	-	-	-	1
8	20	10	-	-	-	1
9	50	25	-	-	-	1
10	100	50	-	-	-	1

Another one-class method is provided by the TF-IDF classifier (TF-IDFC), which is based on the TF-IDF function of 1. The idea behind TF-IDFC is that TF-IDF assigns higher values to the most relevant tokens of a text, thus tweets with higher accumulated TF-IDF scores are more likely to be related with the positive concept defined by the training domain. The proposed unary TF-IDFC relevance score is presented in Table 4.

The last explored unary method is OC-SVM, which has been used for the classification of texts [51]. In this paper, we test two OC-SVM kernels: linear and Gaussian. Both models contain the $\nu \in [0, 1]$ hyperparameter, a lower bound for the number of samples that are support vectors and an upper bound for the number of samples that are on the wrong side of the hyperplane. The Gaussian kernel as the γ hyperparameter that controls the bias-variance trade-off. In this paper, the hyperparameteres were ranged using $\nu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and $\gamma \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1\}$.

3.2.2 Two-class methods for Twitter Financial Disambiguation (TFD)

The binary methods assume a training data with both positive and negative texts ($P \cup N$). Similarly to the unary case, all two-class training methods produce a TFD relevance score (S_t) and a text t is considered positive (value of 1) if $S_t > T_{\text{TFD}}$. The extra negative (generic news) texts allow an adaptation of the TF-IDFC method, as defined in Table 4. Moreover, binary texts enable the computation of other information measures, namely IG and PMI, which are popular in text mining tasks [54, 55]. Following the formulation reported in [55], for each token i of a text t, IG is computed as:

$$\begin{aligned} IG(i) = p(i,1)\log\frac{p(i,1)}{p(i)p(1)} + p(\bar{i},0)\log\frac{p(\bar{i},0)}{p(\bar{i})p(0)} \\ - p(\bar{i},1)\log\frac{p(\bar{i},1)}{p(\bar{i})p(1))} - p(i,0)\log\frac{p(i,0)}{p(i)p(0)} \end{aligned} \tag{2}$$

where the probabilities $p(i), p(\bar{i}), p(1), p(0), p(i,1), p(\bar{i},0), p(\bar{i},1)$ and $p(i,0)$ are derived from the training set ($P \cup N$) and $\bar{i}$ refers to the absence of i. The PMI measures the probability of word co-occurrence in a corpus as:

$$PMI(i,y) = \log\frac{p(i,y)}{p(i)p(y)} \tag{3}$$

where $p(y)$ is the probability of occurrence of class $y \in \{0,1\}$ in the set of training documents (corpus). The adapted IG and PMI TFD relevance scores are shown in Table 4.

Having access to two-class labeled texts also enables the training of supervised ML algorithms. In this paper, we compare three modern classifiers [56, 57]: RF, SVM and a MLP. RF is an ensemble method that combines N_T decision trees based on bagging and random selection of input features. SVM are widely used in text classification [58], computing the best separating hyperplane in a feature space, which is defined by a kernel transformation. The model includes the C hyperparameter, which controls the trade-off between fitting the errors and obtaining a smooth decision boundary. The adopted MLP, also known as Deep Feedforward Neural Network (DFFN), includes [57]: the ReLU activation function on all hidden units (with the sizes h_1, h_2 and h_3), the logistic function on the output layer, a dropout regularization of 0.3 and early stopping (to reduce overfitting). Since the TFD task is unbalanced, a undersampling procedure was applied to the ML training data, which reduces the computational cost when compared with oversampling [59]. Although the training sets are balanced, the test data (from Twitter) is kept with the original unbalanced distribution. The ML algorithms were implemented by using the `keras` and `sklearn` Python modules. The tested hyperparameters include: RF – $N_T \in \{10, 50, 100, 250, 500, 1000, 1500, 3000, 5000, 10000\}$; SVM – $C \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}$, linear and Gaussian kernel ($\gamma \in 0.001, 0.01, 0.05, 0.1, 0.5, 1\}$); MLP – ten different MLP structures related with different combinations of number of hidden nodes, as detailed in Table 6.

The adopted ML binary classifiers (RF, SVM and MLP) output a relevance class probability that can be interpreted as the relevance score $S_t = p(t)$, where $p(t) \in [0,1]$ denotes the class probability for text t. In terms of input variables,

we tested three different types of setups: TF-IDF, W2V or TFD features. TF-IDF and W2V are described in Section 3. The last setup is based on TFD binary statistical measures (TF-IDFC, IG and PMI scores, as computed in Table 4) and, as proposed in [60], k topic relevance features, as obtained using both LDA and BTM text clustering algorithms. Thus, the number of inputs for the TFD features setup is $3+2k$ (α_k values for LDA and θ_k values for BTM). To set k, we apply the Griffiths test [61] on the sample of binary texts when searching for $k \in \{2, ..., 100\}$.

Table 6 Different MLP structures compared.

Network number	Hidden layer size (h1)	Hidden layer size (h2)	Hidden layer size (h3)
1	50	25	10
2	100	50	25
3	100	25	5
4	150	100	20
5	150	50	10
6	200	100	50
7	250	200	20
8	300	150	10
9	500	250	50
10	500	100	5

3.2.3 Financial users relevance rank (FUR)

By using a TFD model, the keywords query resulting texts (Q) can be assigned with a financial relevance score $S_t, \forall t \in Q$. Let Q_u denote the subset of Q texts written by user $u \in U$, where U represents the full set of users that have written the retrieved Q texts. The aggregated FUR score is obtained by summing or averaging all user u texts, where $FUR_u = \sum_{t \in Q_u} S_t$ (sum) or $FUR_u = \frac{\sum_{t \in Q_u} S_t}{|Q_u|}$ (mean).

Similarly to the TFD classification case (Section 3.2.1), a FUR user binary classification can be achieved by adopting a T_{FUR} a decision threshold, which can range through any $FUR_{u \in U}$ domain value. If $FUR_u > T_{\text{FUR}}$ then user u is classified as relevant (value of 1) for the specific financial application, else it is considered as irrelevant (value of 0).

3.3 Evaluation

All evaluation metrics were computed using the Python `sklearn` module. The TFD models are validated by adopting the realistic rolling window procedure (Figure 2) [40,3]. This procedure simulates several training and test model iterations through time (total of I iterations), thus preserving the time order of the news titles and tweets. A fixed time period is used to dimension the training (t_{train}) and test window (t_{test}) texts. In the first iteration, the oldest news titles data are used to train the classifiers. Then, TFD predictions are performed over a Twitter test set, with more recent data. In the second iteration, both the training (news titles) and test (tweets) sets are updated by discarding the oldest texts and adding more recent ones, allowing to train new classifiers and obtain new TFD tweet predictions, an so on. Using the same procedure of [3], to get an overall classification performance we average all I iteration predictive performance metrics. Then, we apply the non-parametric Wilcoxon test for measuring statistical significance [62].

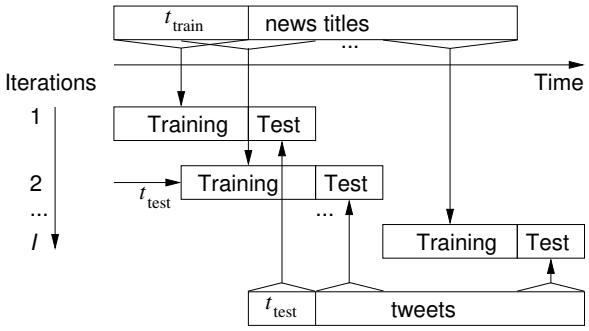


Fig. 2 Schematic of the rolling window procedure.

To compare the different classifiers, we use the popular area under the curve (AUC) of the receiver operating characteristic (ROC) curve [45,63,64], computed on the rolling window test data. The ROC curve shows the performance of a classifier for a target class and across all decision threshold values (T_{TFD} and T_{FUR}), plotting the False Positive Rate (FPR), in x-axis, versus the True Positive Rate (TPR), in the y-axis. The $AUC = \int ROC dT$ measures the global discriminatory performance of a classifier. Often, the AUC values are interpreted as [64]: 0.5 – equal to a random classifier; 0.6 – reasonable, 0.7 – good; 0.8 – very good; 0.9 – excellent; and 1 – perfect. The ROC curve analysis contains two main advantages to evaluate classifiers [63]. First, it is not dependent on the class frequency, thus it can be applied to unbalanced tasks that often occur in text classification, such as the alloy steel TFD. Second, it is not dependent on a specific decision threshold value, which corresponds to a particular TPR (sensitivity) versus FPR (one minus the sensitivity) trade-off.

4 Experimental evaluation

4.1 Data

The Twitter data were collected from March 2017 to October 2018, using the API service and the `Rtwitter` R tool package. The tweets are written in English and related to the following keywords: *steel price*, *steel industry* and *steel production*. A total of 533,759 tweets were retrieved, related with 270,613 unique users.

Since the collected unlabeled Twitter dataset is quite large, we executed a manual labeling of randomly sampled tweets and users to set the ground truth to validate the TFD and FUR models. We created two sets of binary labeled tweets, with 11,081 and 3,000 texts each. The first set is used to tune the TFD model hyperparameters, thus it can be also viewed as a validation set, and to compare the diverse TFD models. The second set is used as an external test set, to estimate the generalization capabilities of the best TFD models on a different unseen dataset. We note that these tweets are unbalanced, presenting an average around 36% of positive texts. Regarding the Twitter user ground truth, we first filtered users that have at least one non-retweet message. Recently, the steel sector received an increased news coverage due to tariffs imposed by the US Government.

As a consequence, many users retweeted steel news just for political reasons, thus the filter allowed to discard a large portion of such users, resulting in 52,653 user accounts. From this set, we randomly selected 418 users that were manually labeled as relevant (1) or irrelevant (0) for the alloy steel domain. The user ground truth set is smaller than the labeled tweets since the manual inspection of a user (e.g., historical tweets, user profile metadata, user web pages) requires much more effort when compared with a single tweet analysis.

To build the training labeled data, we adopted news titles for two main reasons. First, the titles are freely available and easy to collect, while the full news content requires the payment of a fee, specially for steel news media. Second, the length of a title is shorter than the news, thus being closer to the tweet size. The P positive texts were collected from authoritative steel news media: *Kallanish Commodities*[4] and *SteelOrbis*[5]. The news titles are related to the same period of tweets, thus from March 2017 to October 2018. The total number of news titles are 20,366 from *Kallanish Commodities* and 9,418 from *StellOrbis*. Regarding the N negative texts, we used three different generic news sources: 2,554 titles from *The New York Times*[6], 2,990 titles from *Reuters*[7] and 44,182 from the dataset built in [65][8]. The generic news texts are related to the same time period of the collected tweets and steel news. The news titles and the 3,000 labeled tweets are publicly made available[9].

4.2 TFD results

For the TFD model experiments, we adopted a rolling window with a fixed training window size of $t_{\text{train}} = 2$ months and test window of $t_{\text{test}} = 1$ month, which results in a total of $I = 18$ iterations (Twitter test data from May 2017 to October 2018). In the first set of experiments, the overall rolling window test data is composed of the 11,081 labeled tweets. Diverse one-class and two-class TFD models were compared, using different token handling (as detailed in Table 4) and input setups (for the binary ML methods described in Section 3.2.2).

Several of the TFD models include parameters (e.g., E for the W2V embedding size, C value of SVM, M maximum number of tokens). Both tweets and news titles were first preprocessed (e.g., punctuation and stopwords removal), resulting in an average size of 7 words for news titles and 14 tokens for tweets. The token truncation value (M), used by the TF-IDF or W2V input ML models, was set to the average text length since preliminary experiments have shown a better performance of average truncation when compared with the max length value. To set the other parameters, a grid search was executed with the ranges described in Section 3. Similarly to the work of [2], to facilitate the comparison and select a single model throughout all rolling window iterations, the best average AUC configuration model was selected, as presented in Table 7. For comparison purposes, the best TFD model for one-class and two-class cases are further compared with

[4] https://kallanish.com/en/

[5] https://www.steelorbis.com/

[6] https://www.nytimes.com/

[7] https://www.reuters.com/

[8] https://www.kaggle.com/therohk/million-headlines/home

[9] https://github.com/paolazola/Twitter-Financial-Disambiguation-Financial-Users-Relevance

three selected baseline approaches. Table 8 shows the respective AUC values with the three baselines: the Lesk WSD algorithm [17], implemented using the `nltk` Python module; the LDA when the number of topics is set equal to two (aiming to distinguish steel alloy texts); and a supervised binary SVM that is trained using labeled Twitter data and a bag of words (BOW) approach (the SVM uses all input words and it is set using the same modeling procedure, namely rolling window with two months of undersample training data and grid search for hyperparameter selection).

Table 7 TFD classification performance using the 11,081 labeled tweets (average AUC values, best results when using the same type of training data are in **bold**).

Training	Model	Token handling/ Input setup	AUC
One-class Steel news titles	CD	W2V ($E = 1$)	0.49
	DTW	W2V ($E = 1$)	0.44
	SiAE (network 0)	W2V ($E = 16$)	0.60
	Deep SiAE (network 9)	W2V ($E = 8$)	0.62
	TF-IDFC	TF-IDF	**0.78***
	OC-SVM (linear kernel, $\nu = 0.1$)	TF-IDF	0.76
Two-class news titles	CD	TF-IDF	0.64
	DTW	W2V ($E = 16$)	0.72
	TF-IDFC	TF-IDF	0.78
	IG	string	0.60
	PMI	string	0.76
	RF ($N_T = 10000$)	W2V ($E = 8$)	0.75
	SVM (linear kernel, $C = 100$)	W2V ($E = 32$)	0.77
	MLP (network 10)	W2V ($E = 16$)	0.78
	RF ($N_T = 50$)	TFD features ($k = 17$)	0.76
	SVM (linear kernel, $C = 0.001$)	TFD features ($k = 17$)	**0.80**$^{\diamond}$
	MLP (network 6)	TFD features ($k = 17$)	0.79

$\star$ – Statistically significant (p-value < 0.05) under a pairwise comparison when compared with the one-class models: CD, DTW, SiAE and Deep SiAE.
$\diamond$ – Statistically significant (p-value < 0.05) under a pairwise comparison when compared with the two-class models: CD, RF ($N_T = 10000$), SVM (linear kernel, $C = 100$) and MLP (network 10)).

Table 8 Comparison of TFD best classification performances with baselines (average AUC values).

Training	Model	Input setup	AUC
One-class	TF-IDFC	TF-IDF	0.78
Two-class	SVM (linear kernel, $C = 0.001$)	TFD features	0.80
Baselines	Lesk WSD	string	0.50
	LDA	string	0.52
	SVM (linear kernel, $C = 0.5$)	BOW	0.91

When analyzing the comparison results (Table 8), it is relevant to note that the unsupervised Lesk WSD method and the unsupervised LDA provide a poor performance (AUC of 0.50 for Lesk and 0.52 for LDA, equivalent to a random classifier) and that is clearly outperformed by most TFD models. Overall, the best results (Table 8) are achieved by the Twitter trained SVM model (AUC of 0.91). Yet, this model requires a substantial human effort for labeling data, which is prone to errors and it is often unfeasible in practice (e.g., when analysing

big data). Regarding the transfer learning models (Table 7), the best one-class performance of AUC=0.78 is provided by the TF-IDFC statistical method, which is fast to compute and does not contain hyperparameters. The TF-IDFC model AUC differences are statistically significant when compared with all unary methods except OC-SVM. The second best one-class method is OC-SVM (AUC of 0.76), which uses the same set of TF-IDF input features, followed by the autoencoders (AUC of 0.62 and 0.60). The distance based measures (CD and DTW) achieve the worst one-class performances (lower than random classifier). Turning to the binary methods based on string, TF-IDF or W2V tokens, the best results are obtained by TF-IDFC and MLP with W2V, with an AUC of 0.78, which is equal to the one-class TF-IDFC performance. Several of the other direct token input binary methods achieve an AUC higher than 0.7 (SVM, RF, PMI and DTW). The two-class distance measures (CD with AUC of 0.64 and DTW with AUC of 0.72) obtain a substantial performance improvement when compared with their one-class versions (e.g., there is a 28 percentage point increase for DTW). Overall, the best two-class performance is achieved by the SVM that uses the TFD features as inputs, obtaining a very good discrimination level (AUC of 0.80), which is statistically significant when compared with 5 other binary models, as shown in Table 7. The two-class SVM presents an improvement of 2 percentage points when compared with the best one-class model (TF-IDFC), although such difference is not statistically significant.

For further TFD experiments, we selected three best models: the Twitter trained SVM model (for comparison purposes); and the proposed TF-IDFC and the SVM (linear kernel, C =0.001, fed with TFD features, $k = 17$) classifiers, which were the best one-class and two-class methods of Table 7. A second rolling window procedure was executed, using the same fixed train and test time periods (t_{train} = 2 months and t_{test} = 1 month, 18 iterations). During this execution, we reused the previously trained TF-IDFC and SVM TFD models and performed predictions for all 533,759 collected tweets (labeled and unlabeled). All these predictions were stored, allowing a later filtering of the relevant Twitter predictions, needed to compute the additional TFD (shown next) and FUR (Section 4.3) results.

Figure 3 plots the global ROC curves for the selected TFD models when considering the second extra labeled test set with 3,000 tweets. The global ROC curves were obtained by merging all the predictions from the 18 rolling window iterations into a single test set [63]. When executing this additional predictive test, the proposed news titles two-class SVM obtains a global AUC value (0.71), which corresponds to a good discrimination level. This model presents the same 2 percentage point difference (as in Table 7) when compared with the one-class TF-IDFC method (AUC of 0.69). In particular, the ROC curve comparison of Figure 3 shows that the news titles SVM provides better TPR values when FPR is low (higher specificity trade-off region) and a very similar TPR results when FPR is high (higher sensitivity area). While the Twitter trained SVM achieves the best results, this model is less useful in practice, since it requires a costly human effort to label the data (as previously discussed). Nevertheless, the comparison results attest the quality of the proposed transfer learning TFD models (e.g., difference of just 9 percentage points).

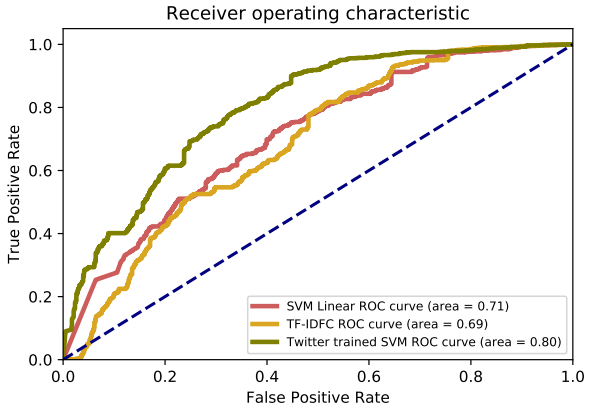


Fig. 3 Global TFD ROC curves and AUC values when using the test sample of 3,000 labeled tweets (dashed line denotes a random classifier).

4.3 FUR results

The FUR experiments used the best TFD models (one-class TF-IDFC and two-class SVM) and their predictions when executing the second rolling window procedure (described in Section 4.2). In particular, we filtered the rolling window predictions to include all tweets related with the ground truth set of 418 users, which resulted in TFD S_t scores for 2,893 unlabeled tweets. These predictions were aggregated by each user u, allowing to compute the global FUR_u and respective ROC curves (Figure 4).

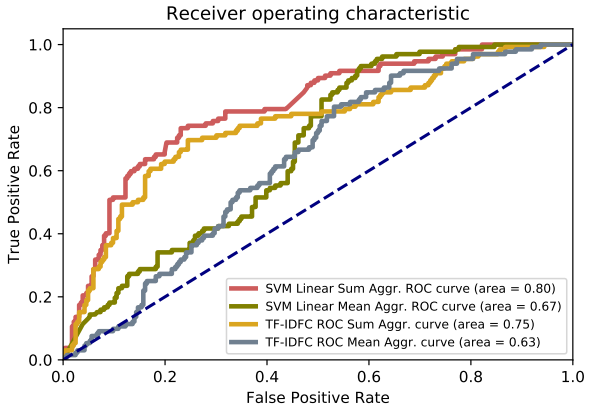


Fig. 4 Global FUR ROC curves and AUC values when using the test sample of 418 labeled users (dashed line denotes a random classifier).

For both TDF models (SVM and TF-IDFC), the best FUR aggregation function is sum, resulting in higher AUC values (13 percentage point difference for SVM and 12 percentage point difference for TF-IDFC). When using the sum aggrega-

tion, the best FUR ROC curve is obtained by the binary SVM model, showing improved TPR values when compared with the unary TF-IDFC for most of the FPR axis range. Overall, the SVM model produced a very good discrimination, presenting an AUC of 0.80 and that is 5 percentage points better than the AUC value of TF-IDFC. It should be noted that the SVM user relevance predictive performance is similar to the one achieved by [45], whose best model provided an AUC of 0.81. However, the authors considered a different Twitter dataset, a different notion of user relevance (not related with alloy steel), and more importantly, used all user history tweets (which requires more memory and computation). In contrast, our FUR approach only considers the tweets that resulted from the financial keywords query (Q).

For demonstration purposes, Table 9 reports the top 20 ranked user accounts when considering the two-class SVM and one-class TF-IDFC FUR sum scores. The **User name** column presents the Twitter account name and Web page for public company profiles. Due to privacy issues, the private accounts were anonymized. As for the **Ground truth** column, it presents the manual label result, where 1 denotes an alloy steel price relevant user and 0 an irrelevant one. The SVM and TF-IDFC rankings only differ after the ninth row. Globally, SVM correctly identifies 15 relevant users and TF-IDFC accurately classifies 14 ones.

Table 9 Top 20 steel price relevant users generated by the FUR scores.

User name	SVM rank	TF-IDFC rank	Ground truth	User name	SVM rank	TF-IDFC rank	Ground Truth
scrapindustry `https://www.scrapmonster.com/`	1	1	1	private user #4	13	18	1
aonesteelgroup `http://aonesteelgroup.com/`	2	2	1	private user #5	14	15	1
marketrnest `http://marketresearchnest.com/`	3	3	1	private user #6	15	16	1
trendy_girl_toy	4	4	0	yicaichina `https://yicaiglobal.com/`	16	-	1
sxcoal `http://www.sxcoal.com/`	5	5	1	private user #7	17	20	1
Cakestreamgo*	6	6	0	ywcdeals `https://yeswecoupon.com/`	18	-	0
foodrecipesgo*	7	7	0	private user #8	19	-	1
breakfastchild*	8	8	0	SPGlobalPlatts `https://www.spglobal.com/platts/en`	20	-	1
private user #1	9	9	1	private user #9	-	10	0
private user #2	10	11	1	private user #10	-	12	0
private user #3	11	14	1	private user #11	-	13	1
Northernweldarc `http://northern-weldarc.com/`	12	17	1	DTradingAcademy `https://daytradingacademy.com/`	-	19	1

∗ – These three Twitter profiles (probably bots) have the same contents and aim to sell or advertise products.

5 Conclusions

Twitter is becoming a valuable big data source for social media analytics. Focusing on financial stocks or indexes, Twitter messages are easily retrieved by using search queries with specific casthags (e.g., $AAPL for Apple stocks). However, the Twitter extraction of other financial opinion tweets, such as related with alloys (e.g., steel, bronze) or commodities (e.g., gold, coffee), is a non-trivial task, as it requires a keywords search that often results in irrelevant texts.

In this paper, we propose an automatic filter approach, termed Twitter financial disambiguation (TFD), aiming to extract financial related tweets and without the need of a human labeling. We achieve this by using a transfer learning approach, in which freely news titles are used to train diverse TFD models, under

two main training approaches: one-class, with only positive texts; and two-class, with positive and negative texts. The TFD models include: adaptations of distance measures (cosine and dynamic time warping); information measures, namely term frequency-inverse document frequency classification (TF-IDFC), information gain (IG) and pointwise mutual information (PMI); and recent machine learning methods, namely simple and deep Siamese autoencoder (SiAE), support vector machine (SVM), random forest (RF) and deep multilayer perceptron (MLP). Also, we test distinct text handling methods, namely the raw string, a TF-IDF transform and a Word2Vec (W2V) encoding. Moreover, given the tweet scores generated by the TFD models, we propose a financial user relevance rank (FUR) score that assigns to each Twitter user a reliability value according to the target financial domain. The advantage of FUR is that it allows to filter relevant users given only the keywords query texts, without the need of additional social media or user features that are typically required by the state of the art studies.

As a case study, we considered the alloy steel prices domain. We performed several steel prices Twitter queries that resulted in 533,759 unlabeled tweets collected from March 2017 to October 2018. Then, we executed a realistic rolling window validation procedure, with several train and test model updates, aiming to tune and compare the diverse one-class and two-class TFD models. The first rolling window experiments, using 11,081 manually labeled tweets as the test set, revealed that the best one-class discrimination performance is obtained by TF-IDFC, while the best two-class training method was obtained by a SVM fed with TFD binary statistical measures (TF-IDFC, IG and PMI) and topic relevance features obtained using the latent Dirichlet allocation (LDA) and biterm topic model (BTM) text clustering algorithms. Overall, the two-class trained SVM model obtained an area under the receiver operating characteristic curve (AUC) of 80%, while the one-class TF-IDFC achieved a slight lower value (AUC of 78%). Both approaches outperformed the Lesk state-of-the-art word sense disambiguation (WSD) method. The two selected transfer learning models were selected for further experiments that used a second rolling window procedure. The experiments confirmed that SVM produces a better discrimination for TFD prediction when using an extra (unseen) set of 3,000 labeled tweets (the AUC was 71% for SVM and 0.69% for TF-IDFC). Moreover, the same rolling window experiment was used to test the SVM and TF-IDFC TFD models predictive performance to discriminate relevant users when using the FUR score and a manually labeled set of 418 users. The best predictive performance was also obtained by SVM, which presented an AUC of 80%, while TF-IDFC obtained an AUC of 75%. In particular, the SVM global Receiver Operating Characteristic (ROC) curve presented better True Positive Rate (TPR) values for most of the False Positive Rate (FPR) axis range. Given these results, we recommend the usage of the two-class SVM model for TFD-FUR, since it consistently provided the best results. As an alternative, in particular if labeled negative tests are not easy to collect, we suggest the simpler one-class TF-IDFC, which does not contain hyperparameters and is faster to compute.

The proposed approach, based on freely labeled news titles, allows an automatic TFD-FUR for Twitter, alleviating the need for a laborious human labeling of tweets or curated lists of relevant user accounts (e.g., web companies) regarding a specific financial domain. Thus, it is valuable as filtering step to be used by financial social media analytics (e.g., sentiment analysis, recommendation users to follow). In terms of limitations, this study only considered data from one appli-

cation domain (alloy steel prices). Also, the proposed two-class SVM algorithm requires a higher computational effort than the simpler one-class TF-IDFC, which becomes a relevant issue when big data is analyzed. In addition, the proposed FUR models were not compared with other user relevance methods. In future work, we intend address these limitations by: considering other case studies, such as commodity (e.g., gold, coffee) or other alloy (e.g., bronze, copper) prices; adapting the proposed two-class SVM algorithm to make use of a more efficient cloud computing infrastructure [66], thus making it more suitable to learn from larger datasets; and comparing FUR with user relevance models that require additional features, such as user account profile data (e.g., web site).

Acknowledgements

Research carried out with the support of resources of Big and Open Data Innovation Laboratory (BODaI-Lab), University of Brescia, granted by Fondazione Cariplo and Regione Lombardia. We would also like to thank the anonymous reviewers for their helpful suggestions.

References

1. J. Awwalu, A. A. Bakar, M. R. Yaakub, Hybrid n-gram model using naïve bayes for classification of political sentiments on twitter, Neural Computing and Applications (2019) 1–14.
2. P. Zola, P. Cortez, M. Carpita, Twitter user geolocation using web country noun searches, Decision Support Systems 120 (2019) 50–59.
3. N. Oliveira, P. Cortez, N. Areal, The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices, Expert Syst. Appl. 73 (2017) 125–144.
4. A. Groß-Klußmann, S. König, M. Ebner, Buzzwords build Momentum: Global Financial Twitter Sentiment and the Aggregate Stock Market, Expert Syst. Appl. 136 (1) (2019) 171–186.
5. V. S. Pagolu, K. N. Reddy, G. Panda, B. Majhi, Sentiment analysis of twitter data for predicting stock market movements, in: 2016 international conference on signal processing, communication, power and embedded system (SCOPES), IEEE, 2016, pp. 1345–1350.
6. F. Lechthaler, L. Leinert, Moody oil: What is driving the crude oil price?, Empirical Economics (2012) 1–32.
7. J. Li, Z. Xu, L. Yu, L. Tang, Forecasting oil price trends with sentiment of online news articles, Procedia Computer Science 91 (2016) 1081–1087.
8. J. Bollen, H. Mao, X. Zeng, Twitter mood predicts the stock market, Journal of computational science 2 (1) (2011) 1–8.
9. S. Feuerriegel, D. Neumann, News or noise? how news drives commodity prices, in: Proceedings of the International Conference on Information Systems, ICIS, Milano, Italy, December 15-18, 2013.
10. T. Rao, S. Srivastava, Modeling movements in oil, gold, forex and market indices using search volume index and twitter sentiments, in: Proceedings of the 5th Annual ACM Web Science Conference, ACM, 2013, pp. 336–345.
11. N. Pröllochs, S. Feuerriegel, D. Neumann, Enhancing sentiment analysis of financial news by detecting negation scopes, in: 2015 48th Hawaii International Conference on System Sciences, IEEE, 2015, pp. 959–968.
12. T. H. Nguyen, K. Shirai, J. Velcin, Sentiment analysis on social media for stock movement prediction, Expert Syst. Appl. 42 (24) (2015) 9603–9611.
13. M. Daniel, R. F. Neves, N. Horta, Company event popularity for financial markets using twitter and sentiment analysis, Expert Syst. Appl. 71 (2017) 111–124.

14. S. Maslyuk-Escobedo, K. Rotaru, A. Dokumentov, News sentiment and jumps in energy spot and futures markets, Pacific-Basin Finance Journal 45 (2017) 186–210.
15. D. Huang, H. Lehkonen, K. Pukthuanthong, G. Zhou, Sentiment across asset markets, Available at SSRN 3185140, 2018.doi:https://doi.org/10.2139/ssrn.3185140.
16. A. Mudinas, D. Zhang, M. Levene, Market trend prediction using sentiment analysis: lessons learned and paths forward, CoRR abs/1903.05440, 2019. arXiv:1903.05440.
17. S. Banerjee, T. Pedersen, An adapted lesk algorithm for word sense disambiguation using wordnet, in: International conference on intelligent text processing and computational linguistics, Springer, 2002, pp. 136–145.
18. P. Zola, M. Carpita, Forecasting the steel product prices with the arima model., Statistica & Applicazioni 14 (1).
19. W. Wei, X. Xia, M. Wozniak, X. Fan, R. Damaševičius, Y. Li, Multi-sink distributed power control algorithm for cyber-physical-systems in coal mine tunnels, Computer Networks 161 (2019) 210–219.
20. C. Lee, J. Won, E.-B. Lee, Method for predicting raw material prices for product production over long periods, Journal of Construction Engineering and Management 145 (1) (2019) 05018017.
21. W. Wei, H. Song, W. Li, P. Shen, A. Vasilakos, Gradient-driven parking navigation using a continuous information potential field based on wireless sensor network, Information Sciences 408 (2017) 100–114.
22. S. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (10) (2010) 1345–1359.
23. X. Liu, Y. Zhou, R. Zheng, Sentence similarity based on dynamic time warping, in: Proceedings of the First IEEE International Conference on Semantic Computing (ICSC), Irvine, California, USA, 2007, pp. 250–256.
24. X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 1445–1456.
25. E. Iosif, A. Potamianos, Similarity computation using semantic networks created from web-harvested data, Natural Language Engineering 21 (1) (2015) 49–79.
26. T. Kenter, M. De Rijke, Short text similarity with word embeddings, in: Proceedings of the 24th ACM international on conference on information and knowledge management, ACM, 2015, pp. 1411–1420.
27. Y. Song, D. Roth, Unsupervised sparse vector densification for short text similarity, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2015, pp. 1275–1280.
28. M. D. Lee, B. Pincombe, M. Welsh, An empirical evaluation of models of text document similarity, in: Proceedings of the Annual Meeting of the Cognitive Science Society, 2005, p. 12541259.
29. M.-W. Chang, L.-A. Ratinov, D. Roth, V. Srikumar, Importance of semantic representation: Dataless classification., in: AAAI, Vol. 2, 2008, pp. 830–835.
30. H. Zhang, K. Yang, E. Jacob, Topic level disambiguation for weak queries, CoRR abs/1502.04823, 2015. arXiv:1502.04823.
31. H. Amiri, P. Resnik, J. Boyd-Graber, H. Daumé III, Learning text pair similarity with context-sensitive autoencoders, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2016, pp. 1882–1892.
32. P. Neculoiu, M. Versteegh, M. Rotaru, Learning text similarity with siamese recurrent networks, in: Proceedings of the 1st Workshop on Representation Learning for NLP, 2016, pp. 148–157.
33. K. H. Lim, S. Karunasekera, A. Harwood, Clustop: A clustering-based topic modelling algorithm for twitter using word networks, in: Big Data (Big Data), 2017 IEEE International Conference on, IEEE, 2017, pp. 2009–2018.
34. D. S. Chaplot, R. Salakhutdinov, Knowledge-based word sense disambiguation using topic models, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), 2018, pp. 5062–5069.
35. X. Li, A. Zhang, C. Li, J. Ouyang, Y. Cai, Exploring coherent topics by topic modeling with term weighting, Information Processing & Management 54 (6) (2018) 1345–1358.
36. Y.-S. Lin, J.-Y. Jiang, S.-J. Lee, A similarity measure for text classification and clustering, IEEE Transactions on Knowledge and Data Engineering 26 (7) (2014) 1575–1590.
37. D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (Jan) (2003) 993–1022.

38. A. Sanborn, J. Skryzalin, Deep learning for semantic similarity, CS224d: Deep Learning for Natural Language Processing. Stanford, CA, USA: Stanford University, 2015.
39. P. Zola, P. Cortez, C. Ragno, E. Brentari, Social media cross-source and cross-domain sentiment classification, International Journal of Information Technology & Decision Making 18 (15) (2019) 1469–1499.
40. L. Tashman, Out-of-sample tests of forecasting accuracy: an analysis and review, International Forecasting Journal 16 (4) (2000) 437–450.
41. Y. Yamaguchi, T. Takahashi, T. Amagasa, H. Kitagawa, Turank: Twitter user ranking based on user-tweet graph analysis, in: International Conference on Web Information Systems Engineering, Springer, 2010, pp. 240–253.
42. C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th international conference on World wide web, ACM, 2011, pp. 675–684.
43. A. Pal, S. Counts, Identifying topical authorities in microblogs, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 45–54.
44. D. Gayo-Avello, Nepotistic relationships in twitter and their impact on rank prestige algorithms, Information Processing & Management 49 (6) (2013) 1250–1280.
45. J. Ito, J. Song, H. Toda, Y. Koike, S. Oyama, Assessment of tweet credibility with lda features, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 953–958.
46. P. Cortez, N. Oliveira, J. P. Ferreira, Measuring user influence in financial microblogs: experiments using stocktwits data, in: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, ACM, 2016, p. 23.
47. A. B. Eliacik, N. Erdogan, Influential user weighted sentiment analysis on topic based microblogging community, Expert Syst. Appl. 92 (2018) 403–418.
48. I. Alsmadi, G. K. Hoon, Term weighting scheme for short-text classification: Twitter corpuses, Neural Computing and Applications 31 (8) (2019) 3819–3831.
49. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
50. Z. Wood-Doughty, N. Andrews, M. Dredze, Convolutions are all you need (for classifying character sequences), in: Proceedings of the 4th Workshop on Noisy User-generated Text, NUT@EMNLP 2018, Brussels, Belgium, November, 2018, pp. 208–213.
51. L. M. Manevitz, M. Yousef, One-class svms for document classification, Journal of Machine Learning Research 2 (Dec) (2001) 139–154.
52. P. Senin, Dynamic time warping algorithm review, Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA 855 (2008) 1–23.
53. L. V. Utkin, V. S. Zaborovsky, A. A. Lukashin, S. G. Popov, A. V. Podolskaja, A siamese autoencoder preserving distances for anomaly detection in multi-robot systems, in: 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), IEEE, 2017, pp. 39–44.
54. Y. Xu, G. J. Jones, J. Li, B. Wang, C. Sun, A study on mutual information-based feature selection for text categorization, Journal of Computational Information Systems 3 (3) (2007) 1007–1012.
55. N. Oliveira, P. Cortez, N. Areal, Stock market sentiment lexicon acquisition using microblogging data and statistical measures, Decision Support Systems 85 (2016) 62–73.
56. T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer-Verlag, NY, USA, 2008.
57. I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.
58. J. Costa, C. Silva, M. Antunes, B. Ribeiro, Boosting dynamic ensembles performance in twitter, Neural Computing and Applications (2019) 1–13.
59. G. E. Batista, R. C. Prati, M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD explorations newsletter 6 (1) (2004) 20–29.
60. J. Cai, W. S. Lee, Y. W. Teh, Improving word sense disambiguation using topic features, in: EMNLP-CoNLL 2007, Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, Czech Republic, 2007, pp. 1015–1023.
61. T. L. Griffiths, M. Steyvers, Finding scientific topics, Proceedings of the National academy of Sciences 101 (suppl 1) (2004) 5228–5235.

62. M. Hollander, D. A. Wolfe, Nonparametric statistical methods, Wiley-Interscience, 1999.
63. T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.
64. S. Gonçalves, P. Cortez, S. Moro, A deep learning classifier for sentence classification in biomedical and computer science abstracts, Neural Computing and Applications (2019). URL https://doi.org/10.1007/s00521-019-04334-2
65. R. Kulkarni, A million news headlines, Tech. rep., Harvard Dataverse, V2 (2018). URL https://doi.org/10.7910/DVN/SYBGZL
66. Wei Wei, X. Fan, H. Song, X. Fan, J. Yang, Imperfect information dynamic stackelberg game based resource allocation using hidden markov for cloud computing, IEEE Trans. Serv. Comput. 11 (1) (2018) 78–89. doi:10.1109/TSC.2016.2528246.