LINGUISTIC FEATURE CLASSIFYING AND TRACING

***Mohammadreza Moohebat*[1], *Ram Gopal Raj*[2], *Dirk Thorleuchter*[3] *and Sameem Binti Abdul Kareem*[4]**

[1, 2, 4] University of Malaya, Department of Artificial Intelligence, Faculty of Computer Science & Information Technology, Kuala Lumpur, Malaysia
[3] Fraunhofer INT, D-53879 Euskirchen, Appelsgarten 2, Germany

E-mail: moohebat@gmail.com[1], ramdr@um.edu.my[2], sameem@um.edu.my[4]

ABSTRACT

We investigate the identification and analysis of linguistic (lexico-grammatical) features that are characteristically used by articles of a specific year of publication. Linguistic features differ from shallow features because they represent authors' lexico-grammatical writing styles and do not consider well-known bag-of-words model. Current literature focusses on shallow features rather than on linguistic features and existing methods for identifying linguistic features use well-known knowledge-structure based approaches. In contrast to this, we advance these existing methods by applying semantic clustering instead of using knowledge-structure based approaches. For evaluation purpose, a linguistic feature-based prediction model is built to enable an automated assignment of articles to their years of publication. In a case study, the proposed methodology is applied to articles of the Springer book series 'Communications in Computer and Information Science' published from 2009 to 2013. The Case study results show the feasibility of the proposed approach as compared to frequently used baseline.

Keywords: Scientific articles, Linguistic features, Latent semantic indexing, Text Mining.

1.0. INTRODUCTION

We investigate the occurrence of linguistic (lexico-grammatical) features in articles to show that they can be used for assigning articles to their years of publication. The Literature shows related approaches that can be used to assign articles to a pre-defined class. A domain- specific vocabulary (key words) is often used for this classification task. Different domains can be well distinguished by the distribution of specific key words as shown by existing bag-of-words approaches [1]–[5]. Further, trend analysis and bibliometric research also show that key word distributions can be used to identify a time period [6]. They trace topic changes over time within a domain. Thus, these approaches can estimate an article's publication year based on the used topics.

The approaches as mentioned above are based on shallow (bag-of-words) features. They are in contrast to linguistic features such as specific word class distributions that indicate authors' lexico-grammatical writing styles. Literature also shows the possibilities of using linguistic features for classification. [7] investigate the impact of linguistic features on different scientific disciplines and on different points in time. A further approach uses linguistic features for spam detection [8]. Both approaches are based on systemic functional linguistics, in which a knowledge-structure based classifier (e.g. support vector machine) is used.

We provide a new approach that identifies articles' linguistic features and that investigates their usage at different points in time. In contrast to previous work, clustering is used instead of classification. Text classification assigns a text to the given pre-defined classes. Classes are normally defined in a way that they cover all known linguistic features that are expected to occur within the given texts. Text clustering automatically identifies new classes from a given text. Thus, the proposed approach is able to identify new, unknown linguistic features from text.

A second contrast to previous work is that semantic clustering is applied instead of using well-known knowledge-structure based approaches. Knowledge-structure based approaches focus on aspects of words. They identify two text patterns as similar if the terms of the patterns are similar. Semantic clustering enables the identification of a similarity among two text patterns, even if they do not share a common term. Thus, semantic approaches identify terms that co-occur together with the terms of the two text patterns. The relationship between these co-occurring groups of terms is used to estimate the similarity of a text pattern. If two (e.g., equally written) terms can be distinguished based on their part-of-speech, then semantic approaches identify terms that frequently co-occur with the first term and the second term separately. This allows the identification of linguistic features.

A disadvantage of clustering approaches is that the classes are built automatically and not manually. Thus, some classes represent linguistic features; however, other classes might represent shallow features. To calculate the impact of linguistic features on the prediction of articles' publication years, shallow features should be strictly separated from the linguistic features. Otherwise, shallow features' impacts can influence linguistic features' impacts. This fact is considered in the proposed methodology by introducing and applying a new feature-based summarizing procedure. For each feature, the impacted terms with the same word class are aggregated. A distribution of word classes is calculated to represent the features. Features with similar word class distributions are summarized. While the new procedure considers only word classes instead of single words, the class reduction keeps linguistic features and discards shallow features.

The well-known latent semantic indexing (LSI) method is applied together with singular value decomposition (SVD) for semantic clustering. SVD dimensions (named features) are created based on training data. Features that represent linguistic features - in contrast to shallow features - are selected. Test data are projected into the latent semantic feature space as created during training. Based on the impact of articles on the linguistic features, predictive modeling is used for evaluation purposes.

The proposed methodology is applied in a case study on articles from Springer book series (Communications in Computer and Information Science). Features are extracted from 2011 articles based on their discriminatory power to features from 2009, 2010, 2012, and 2013 articles. Shallow features are discarded and the linguistic features are used for prediction modeling. As a result, evaluation shows that the proposed methodology outperforms frequent baseline. Thus, it is feasible to use linguistic features for predicting articles' publication year.

Section 2 gives an introduction in LSI and in linguistic research for background purposes. The methodology is depicted in Section 3 and the different steps of the methodology are described in detail in the corresponding sub-sections. Section 4 describes the case study. A detailed evaluation including example results is provided in Section 5. Section 6 concludes the paper.

2.0. BACKGROUND

2.1 Linguistic research

A well-known approach for text analysis in linguistics is Bag-of-Words (BoW). It considers occurrence or non-occurrence of a word in a context where occurrence is indicated by a word frequency, by a weight (tf-idf), by a Boolean value, or by a normalized frequency [9]. Despite the benefits of BoW, it suffers from the problem that text files consist of a huge amount of words. This leads to a huge dimension of the corresponding term vectors and applying a dimension reduction procedure is necessary to perform BoW [10].

With improving the natural language processing (NLP), linguistic methods and theories amalgamated with the text mining context. Extracting the role of the words in the sentences is the first thing that scientists note and implement. Liu (2006) introduces Part-of-Speech (PoS) tagging as "a linguistic category that is defined by its syntactic or morphological behavior". Because PoS tagging is the simplest and earliest form of grammatical annotation, with countless applications in NLP, many different techniques for PoS tagging are proposed. For instance, Cutting et al. (1992) and Kupiec (1992) employ ~~a~~ Markov model. Ratnaparkhi (1996) applies the maximum entropy technique and achieved 96.6% accuracy. In most of the cases, training for PoS tagging is done by using The Wall Street Journal Pen Treebank corpus.

In addition to the PoS tagging, the role of the words in the sentence and the relation of these words is also vital and can give additional information about the sentence structure. Parsing refers to producing the parse tree from the sentence. Parsing is one the fundamental parts of NLP and, similar to PoS tagging, it has many applications in syntactic analysis, information extraction, classification, etc.

While shallow features are defined as word distributions based on the BoW approach, linguistic features are defined as the occurrence of a word class distribution [7]. Utilization of shallow features, linguistic features, or a mixture of both are reported. Moohebat et al. (2013) apply shallow features for the classification of ISI-indexed papers from Non-ISI-indexed papers by three different algorithms [14]. [7] test three types of features: shallow features, linguistic features, and a combination of both of them. The objective is to classify different scientific disciplines correctly. They employ systemic functional linguistics (SFL), PoS n-gram, and information density. Shallow features are the 500 most distinctive words calculated by information gain. These features are implemented on two sets of papers from two different decades (1970 to 1980 and 2000 to 2009). In both cases, SVM is utilized for classification. The result shows that in the years 1970 to 1980, shallow features work better. However, for the years 2000 to 2009, a combination of linguistic and shallow features is more promising. They argued that this difference comes from the fact that during that time, disciplines were not distinct from a linguistic view [7].

Bergsma et al. (2012) conduct a study in the area of scientific writing. Their objectives are predicting the gender of the author, guessing whether he or she is a native speaker, and finally guessing if the paper is presented in a workshop or proceeding. They use shallow and linguistic features. Furthermore, they also use three different syntactic feature styles for comparing: Context Free Grammar (CFG), Tree-Submission Grammar (TSG), and Charniak-Johnson (C&J) re-ranking. Results show that shallow and linguistic features outperform the other three syntactic models [15].

Mixing linguistic and shallow features is promising according to Afroz (2012). She succeeds in detecting deceptive writing style with a 97% F-score. She combines linguistic and shallow features with lying detection and authorship features together [16].

2.2 Latent Semantic Indexing

Latent semantic indexing (LSI) is a method for indexing and retrieval. It uses singular value decomposition (SVD) as a mathematical technique from algebra to discover latent, underlying patterns within a collection of unstructured texts [17], [18]. The patterns consist of several terms that are semantically related. The relationships are identified by considering associations between terms that occur in similar contexts as calculated by term co-occurrences [19], [20].

The use of LSI in our work is motivated by three LSI characteristics. First, LSI is able to work with different types of term definitions. A term can be defined as a single word, as a word in stemmed form, or as a word of different part-of-speech. This enables the calculation of term distributions (shallow features) as well as word class distributions (linguistic features) based on semantic relationships. Second, the clustering aspect of LSI is also interesting. It identifies new, unknown term or word class distributions that are latent in the articles. This enables the identification of new linguistic features from articles. Last, LSI calculates the documents' impact on the identified features. This enables tracing of linguistic features over time, assuming that the articles can be assigned to their year of publication.

Related approaches that can be used as a substitute for LSI clustering are PLSI - 'probabilistic latent semantic indexing' [21], NMF - 'non-negative matrix factorization' [22], [23], and LDA - 'latent Dirichlet allocation'[24], [25]. They are of higher conceptual complexity than LSI. Thus, LSI is used in this paper because it is more comprehensible to the reader.

3.0. METHODOLOGY

Fig. 1 shows the proposed methodology. The different steps are explained in the subsections below. A collection of articles is used where all articles are assigned to their corresponding years of publication.

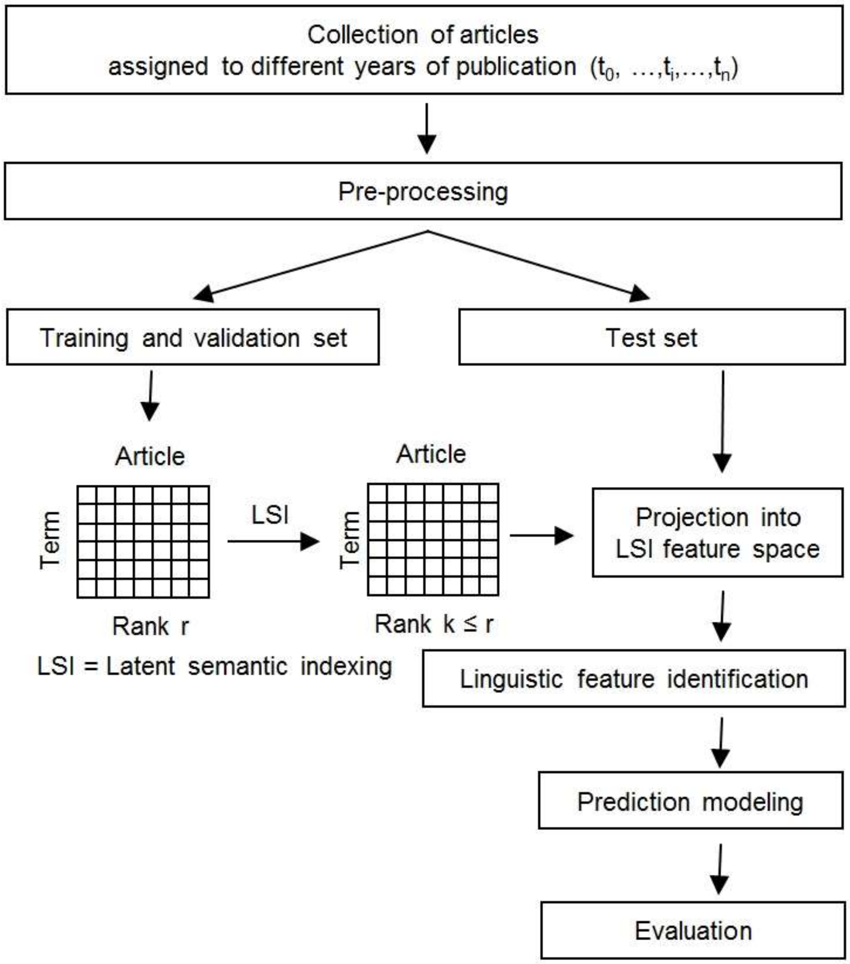
Collection of articles
assigned to different years of publication $(t_0, \ldots, t_i, \ldots, t_n)$
Pre-processing
Training and validation set
Test set
Article
Term
Rank r
LSI
Article
Term
Rank k ≤ r
LSI = Latent semantic indexing
Projection into LSI feature space
Linguistic feature identification
Prediction modeling
Evaluation

Fig. 1: The processing of the proposed methodology in different steps

Articles from a specific year of publication are indicated by a target variable. This enables us to build a binary classification model that automatically assigns new articles for the specific year of publication or alternatively, that assigns new articles to previous / succeeding years. All articles are pre-processed (see Sect. 3.1). Articles are split into training and test set. The training set is used to build the model and to determine its parameters, and the test set is used for evaluation. LSI is applied on the data (see Sect. 3.2) to build an LSI feature space. Articles from the test set are projected into the LSI feature space ~~as~~ created during training (see Sect. 2.2).

This LSI feature space consists of both: linguistic features and shallow features. The aim of this work is to identify linguistic features, to track them over time, and to show their impact on a binary classification model used to assign articles to their years of publication. Thus, linguistic features have to be distinguished from shallow features in a first step. This is done by summarizing features from the feature space concerning the used word classes. By definition, each linguistic feature consists of a unique word class distribution (see Sect. 2.1), where their word class distributions could not be summarized. In contrast to linguistic features, shallow features are characterized by the occurrence of different key words and not by the occurrence of a different word class distribution. It might be that two different shallow features have a similar word class distribution, although they do not share a common meaning. Thus, shallow features are summarized by joining similar word class distribution (see Sect. 3.3). This identifies the linguistic feature standing behind several different shallow features and - together with the linguistic features that are directly identified by LSI - the relevant linguistic features can be identified. In a second step, prediction modeling is applied on the summarized LSI feature space (see Sect. 3.4) to build the binary classification model and to trace the linguistic features. The criteria used for evaluation are discussed in Sect. 3.5.

3.1 Data collection and pre-processing

For data collection, some assumptions have to be considered. Articles should be arranged according to the year of publication. To ensure statistical relevance, a minimum number of articles per year should be provided (e.g. n > 1000). Articles should stem from a specific domain and they should represent the domain by covering domain-specific aspects.

Article collections that are in accordance with the assumptions described above normally differ from the document collections available, e.g. on the internet. This is because the number of articles has been often too low, articles stem from different domains, articles stem from a domain that is not of interest for analyzing, or an assignment of each article for its year of publication is missing. Thus, existing document collections normally cannot be used for this task. Further, manual creation of such an article collection is too time-consuming because of the large number of articles required per year.

Data collection can be performed by retrieving articles from specific databases. Normally, the resulting documents are provided in a specific data format (e.g., pdf, Microsoft Word Document). For further processing, they have to be converted to plain text format. Additionally, each resulting document might contain several articles. Extracting the articles manually is too time-consuming. Thus, an automated procedure has to be created specifically for the data which enables automated extraction of articles from the resulting documents.

The retrieved plain text articles are pre-processed. Xml, html-tags and scripting code are deleted as well as specific characters. To identify single terms, tokenization with term unit as the word is applied. Based on a dictionary, typographical errors are corrected. PoS tagging is used to assign each term to its word class.

In contrast to standard pre-processing procedures from the literature, stop word filtering and Zipf's law are not applied and stemming is applied in a different way. Stop words are terms that occur frequently within each article of the article collection. Their word classes are a part of the corresponding linguistic feature and thus, stop words have to be retained. Stemming groups terms with the same stem. It might be that stemming groups terms of different word classes. This would change the corresponding linguistic feature. To prevent this, stemming is applied to each word class separately. According to the Zipf distribution [26], a term that only occurs once or twice in an article collection is discarded. This is because of its low content value. For analyzing word classes, the content value of a term is not relevant, but the word classes of these low- frequency terms determine the corresponding linguistic feature. Thus, these terms also have to be retained.

For each article, a term vector in the vector space model is built. The vectors' components are distinguished terms from the articles by considering the corresponding word classes, e.g. the term "laser" as word class "noun" occurs as a vector component, and the term "laser" as word class "verb" also occurs as a separate vector component. This enables us to analyze the word class distribution. Vectors' components are represented by weighted term frequencies. This is because a term vector consists of terms from articles of different lengths (with a different number of terms). Raw frequencies are directly impacted by articles' lengths while weighted term frequencies is used as a length normalization factor to decrease this impact.

A well-known weighting scheme is used [27]. The weight $w_{i,j}$ for term i and for article j is calculated by

$$w_{i,j} = \frac{tf_{i,j} \cdot \log(n/df_i)}{\sqrt{\sum_{p=1}^{m} tf_{i,j_p}^{\ 2} \cdot (\log(n/df_{i_p}))^2}} \quad (1)$$

The number of articles is n, the number of articles that contain term i is df_i, and the number of distinguished terms is m. The product of term frequency $tf_{i,j}$ and inverse article frequency $\log(n/df_i)$ is divided by a length normalization factor [28].

The term vectors created above are split into training set and test set. The training set is used to build and train the prediction model. The test set is used for evaluation purposes.

3.2 Latent semantic indexing

The term vectors from the training set are composed to build a term-by-article matrix *A*. Based on the assumptions in Sect. 3.1, a large number of articles is required and thus *n,* the number of articles, is large. Although the articles stem from the same domain, they cover different aspects within the domain and they are written by different authors using different words. Thus, *m* as the number of distinguished terms of the articles is also large. This leads to a large rank *r* of the term-by-article matrix A ($r \leq \min(m, n)$) and it makes the use of A for further evaluations unmanageable. A well-known solution for this problem is to reduce the rank by applying LSI. It identifies k semantic textual patterns (features) that occur latent in several articles and that are of high discriminatory power between each other. While *k* normally is much smaller than *r*, evaluating the results of LSI is manageable [29]. The *k* features are composed to build a new matrix A_k with rank *k* as an approximation of *A*. The formal description of this process is to split matrix *A* into three matrices by use of singular value decomposition as a first step.

$$A = U\,\Sigma\,V^t \quad (2)$$

Σ is used to reduce the rank of A because on its diagonal components the singular values of A are ordered by size ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$). After calculating a value of k<r, matrix Σ_k can be built that contains the first k singular values of A while the other singular values are discarded. U and V are also reduced to U_k and V_k by discarding all columns of U and V from k+1 on. The new matrix A_k as the approximation of A with lower rank k is built in a second step.

$$A \approx A_k = U_k\,\Sigma_k\,V_k^t \quad (3)$$

While Σ_k is not relevant for further processing, the LSI feature space is defined by U_k and V_k. U_k components can be interpreted as the impact of a term on one of the k features. V_k components can be interpreted as the impact of an article on one of the k features. A new article can be projected into this LSI feature space. The term vector v'_d of the new article has to be transformed to a new vector v_d that is in accordance to the structure of matrix V_k [30]. Thus, the k components of v_d can be interpreted as the impact of the new article on the corresponding feature. Deerwester et al. (1990) propose the formulation for that transformation:

$$v_d = v'_d\,U_k\,\Sigma_k^{-1} \quad (4)$$

3.3 Linguistic feature identification

As a result of Sect. 3.2, linguistic features are identified as well as shallow features. Features are summarized concerning the used word classes to identify further linguistic features standing behind shallow features and to discard the shallow features. To summarize the features, a word class distribution is calculated for each feature. The columns of matrix U_k represent the impact of a term for each of the k features. These terms are selected to represent a feature where its impact is above a specific threshold. The word classes of the selected terms are used to calculate the word class distribution among the selected terms. This enables the calculation of the word class distribution for each of the k features. These k word class distributions are compared against each other to identify similar distributions. Comparison is done by applying a standard similarity measure and by determining a specific threshold to define similarity. Two or more features with similar word class distributions are joined. This is done by using the average impact value of these features as a component in matrix U_k.

Feature summarizing also has an impact on matrix V_k. Here, the maximal impact value of all features is used as a component in matrix V_k. An example is an article that has an impact on a first feature of 0.4 and an impact on a second feature of 0.5. In matrix V_k, the component values of the corresponding article row and the corresponding feature columns are 0.4 and 0.5. After joining both features, the two columns are joined as one column by keeping the larger value 0.5 and by discarding the smaller value. The reason for selecting the maximal impact value is to retain articles with high impact on one feature and with lower impact on the other similar features.

Feature summarizing is important for the projection of test data into the LSI feature space. The created vectors v_d are composed to a matrix $V_{k\text{-test}}$. Joining the components of $V_{k\text{-test}}$ shows the impact of articles with a specific publication year on the summarized features. Thus, changes in the use of these linguistic features can be traced over time.

3.4 Prediction modeling

The logistic regression model of training set $T = \{(x_i, y_i)\}$ is shown below. N is the number of articles in the training set and $i = \{1,2,...,N\}$. Further, $x \in R^k$ represents a k-dimensional term vector in LSI feature space and it is calculated in accordance to term vector v_d. Thus, it shows the impact of the article on each of the k features. Further, w is a parameter vector and w_0 the intercept. $y_i \in \{0,1\}$ represents a binary target label for each article. It indicates whether an article can be assigned to a specific publication year or not.

$$P(y = 1 | x) = \frac{1}{1 + \exp(-(w_0 + wx))} \tag{5}$$

3.5 Evaluation criteria

To evaluate the prediction model, we use the cumulative lift, the sensitivity and specificity, and the area under the receiver operation characteristics curve. The lift measure indicates density increases of articles that are successfully assigned to a specific publication year in relation to the density of all articles belonging to the publication year. TP (true positive) is defined as the number of articles correctly assigned to a given publication year and FN (false negative) is defined as the number of articles that are not assigned to the publication year by the model but this assignment is incorrect. TN (true negative) is defined as the number of articles that are correctly not assigned to the publication year and FP (false positive) is defined as the number of articles that are assigned to the publication year by the model but this assignment is incorrect. The sensitivity (TP/(TP+FN)) is defined as the ratio of correctly assigned articles from the publication year to all articles from the publication year. The specificity (TN/(TN+FP)) is defined as the ratio of correctly assigned articles with different publication year to all articles with different publication year [32]–[34]. The two dimensional plot of the sensitivity versus (1-specificity) is named the receiver operation characteristics curve (ROC) [35], [36]. It is used to calculate the AUC (area under the ROC) as measure for comparing performance of binary classification models [37] .

4.0. CASE STUDY

An empirical evaluation of the proposed approach is done in this case study. The aim is to provide a binary classifier for assigning a new article to a given publication year based on linguistic features of training articles and to show its success.

4.1 Data collection and pre-processing

To consider the requirements for data collection, scientific articles are used. They can be assigned to a specific discipline or sub-discipline and thus, to a specific domain. In contrast to news and social media articles, scientific articles are characterized to some degree by a homogeneous writing style and normally, it is expected that this style does not change over a small number of years. Thus, the identification of changes in linguistic feature usage per year may provide valuable insights for linguistic research.

Scientific articles are often composed and published in books or journals. It is easy to gain access to a specific article in pdf format; however, it is hard to obtain a collection of a large number of articles. The publisher Springer provides institutional access to its book series, 'Communications in Computer and Information Science'. Each book contains articles successfully provided at a conference in a specific year. The conferences are from the computer science discipline and they can be further assigned to the sub-discipline: 'Database Management & Information Retrieval'. The conference year is used as the publication year for all articles in the corresponding book. Thus, all books in the series from conferences between 2009 and 2013 are collected. As a result, 357 Springer books are collected that are written in English language.

Each book is provided in pdf format. In a manual process, Adobe PDF Writer is used to open all books and to convert them to plain text format. This discards image in the text, but the text itself and text in tables are retained. While each book contains many articles, a self-developed text mining application is used to extract the articles from books automatically. It identifies the beginning and end of an article and saves each article as a separate text file. Overall, 9510 articles are extracted and used for further processing.

Table 1: Overview on the data characteristics

	Number of articles	Relative percentage
Training set:		
Articles from 2011	2,024	42.54
Articles from 2009, 2010, 2012, 2013	2,734	57.46
Total	4,758	
Test set:		
Articles from 2011	2,013	42.36
Articles from 2009, 2010, 2012, 2013	2,739	57.64
Total	4,752	

SAS 9.3 Textminer is used for pre-processing on each text file. The methods described earlier, especially part-of-speech tagging, is used to assign each term to its word class. As a result, a term vector is created for each text file. SAS 9.3 Textminer also splits about 50% of the term vectors into training sets and about 50% into test sets. The data partition is based on a simple random method applied per conference, but not applied per single article. Thus, articles from the same conference are assigned to the same set (training or test set). This is because some conferences may have specific format requirements and the shallow features representing this requirement only can be found in articles from this conference and vice versa. These shallow features strongly influence classification performance. Assigning all articles from a conference in total to training set or to test set prevents the classification performance from being impacted by these shallow features.

The target value for prediction modeling is the publication year 2011. It is estimated weather an article is from 2011 or not. Thus, positive examples are articles from 2011 and negative examples are articles from 2009, 2010, 2012, and 2013. Data characteristics are shown in Table 1. As depicted, 42.36 % of test set articles are from 2011. This value is used as a frequent baseline for the evaluation.

4.2 Linguistic feature identification

SAS 9.3 Textminer is used for processing LSI on the training data. It enables to calculate an optimal value of k as a compromise between the number of features (k) and the overall discriminatory power of the features compared to each other. As a result, k is set to 70 by LSI. For a further reduction, feature summarization according to Sect. 3.3 is applied: A word class distribution is built for each feature. It consists of word classes and their occurrence probabilities from terms with high impact on the corresponding feature. Impacts i are taken over from the corresponding impact values of U_k ($i \in [-1,..,1]$). To determine high impact terms, a threshold (i>+0.6) is selected. Simple Euclidean distance between the features is used together with a further threshold (80% similarity) to identify related features. The reason of using Simple Euclidean distance was its natural simplicity. As a result, 40 linguistic features are identified.

Table 2: Characteristics of selected linguistic features: occurrences of word classes in percentage

Linguistic feature:	1	2	3	4	5
Abbreviation	0.53	0.31	0.19	1.28	0.24
Adjective	16.06	15.11	17.42	10.97	21.71
Adverb	2.09	1.98	2.30	0.54	1.14
Auxiliary	0.00	0.26	0.00	0.00	0.01
Common Noun	58.68	55.17	55.72	45.19	54.50
Conjunction	0.97	0.53	0.09	0.00	0.02
Determiner	0.00	1.19	0.00	0.00	0.24
Interjection	0.04	0.1	0.03	0.01	0.03
Preposition	0.13	1.32	0.52	0.05	0.56
Pronouns	0.00	0.22	0.05	0.68	0.53
Proper Noun	17.10	15.82	18.87	37.88	15.73
Verb	4.40	7.99	4.81	3.40	5.29

Table 2 shows the characteristics of five selected linguistic features. At a first glance, the word class distributions of the features look similar. Common noun is most frequently used word class followed by proper noun and adjective. However, a detailed view clearly indicates differences in word class distribution between the features. These differences are used for prediction modeling. Table 3 gives an overview of terms standing behind the different word classes. It is important to know that the word class distribution is built based on terms with high impact on a corresponding linguistic feature. Stop words are words that occur frequently in all articles of a collection as indicated by a low term weight. Thus, they occur homogeneously in all linguistic features and they normally could not be used to characterize a specific linguistic feature. Therefore, LSI assigns a low impact value to these stop words. While the word class distribution is built based on high impact terms, stop words are not considered. Auxiliaries, conjunctions, determiners (e.g. article, demonstrative, possessive), interjections, preposition, and pronouns normally consist of these stop words. This is the reason why the percentage of these word classes is low in the word class distribution of all linguistic features.

Table 3: Examples for terms in the provided articles with high impact on at least one feature and their assignment to word classes

Word class	Terms
Abbreviation	uhf, vhf, isbn, app, admin, mgmt, et al., ft, kg, mol, vs, rd
Adjective	fuzzy, international, semantic, complex, similar, specific, average
Adverb	respectively, effectively, relatively, currently, even, specifically gradually, approximately
Auxiliary	ought to, used to, would, could, need, will, may, can, need to, dare to, should, might, cannot, must, shall, dare, does, needs to
Common Noun	information, model, process, network, algorithm, node, process, time, application, results, analysis, value, problem, software, technology, set, performance, knowledge
Conjunction	as well as, so that, whereas, even if, however, even though, either, in case, whenever, as long as, assuming that, as soon as, given that, considering that, provided that, as if
Determiner	few, her, my, whatever, your
Interjection	thank you, yes, so long, so
Preposition	according to, during, among, due to, including, below, with respect to, instead of, upon, up to, in addition to, near, in accordance with
Pronouns	each, how much, whom, someone, everyone, anything, nothing, anyone, neither, no one, ourselves, everybody, myself, anybody
Proper Noun	wang, zhang, springer, chen, acm, mobile, university, table, xml, lncs, proc, figure, yang, beijing, huang, new york, web, case, system
Verb	Propose, represent, follow, take, obtain, proposed, find, apply, set, improve, develop, define, get, include, generate, see, require, provide, compare

4.3 Prediction modeling

A regression model is built that assigns training articles to their publication year (binary classification: 2011 or non-2011) based on the identified 40 linguistic features. Test articles are projected into the LSI feature space. The identified 40 linguistic features are selected while other features are discarded. Based on the selected features, the regression model predicts test articles to a publication year 2011 or to non-2011. An evaluation is done to show the predictive performance of the regression model and to compare them to the frequent baseline.

Fig. 2 shows the cumulative lift curve of the test set. It is situated above the baseline. Within a specific percentile, the test set is able to assign more articles correctly to its year of publication than the baseline. The ROC curve of the test set is depicted in Fig. 3. It also lies above the frequent baseline. This performance improvement is significant (χ^2=0.02, d.f.=1, p<0.001) because AUC of the test set (0.6990) is larger than the baseline (0.5000). Thus, the model is able to better distinguish articles from 2011 from articles that are not from 2011 than the baseline.

.

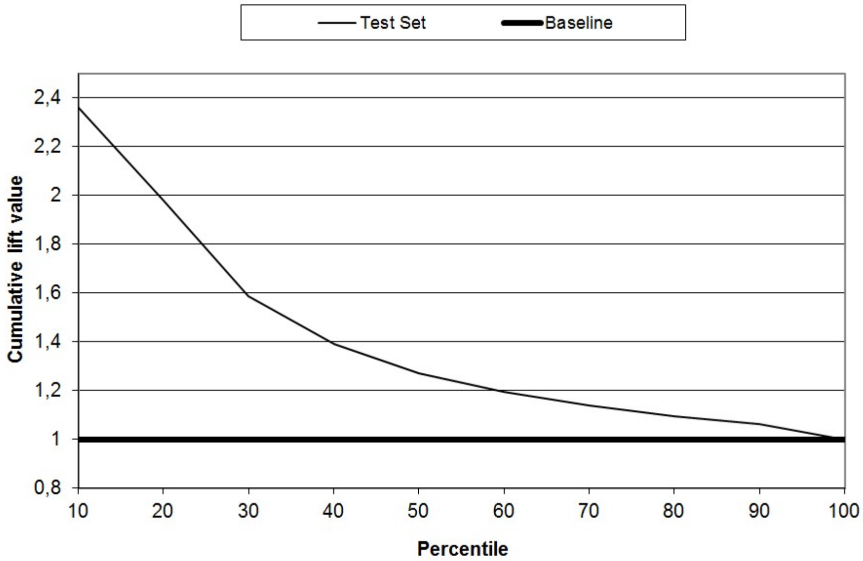

Fig. 2: Cumulative lift of test set and baseline for the logistic regression model

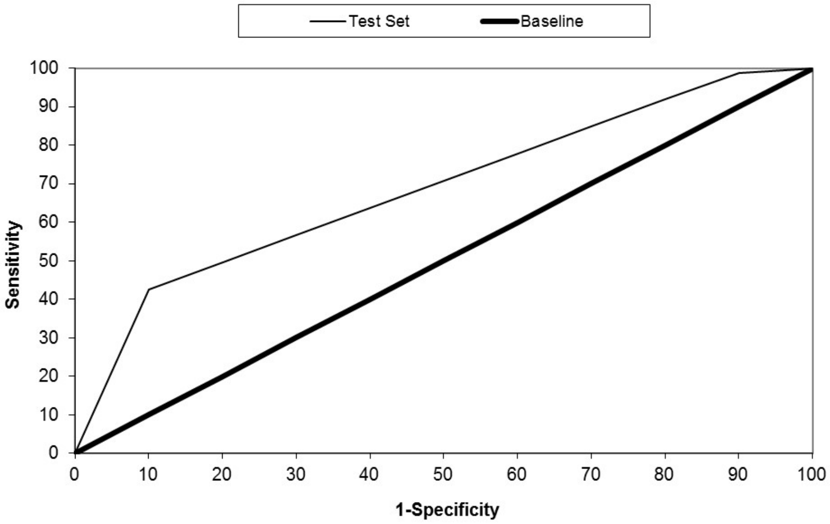

Fig.-3: Sensitivity / Specificity Diagram

Overall, results show that using linguistic features for assigning articles to their publication year is feasible because case study results outperform frequent baseline.

5.0 CONCLUSION

A new approach is provided that identifies linguistic features from articles. In contrast to related work, it uses clustering instead of classification and it uses semantic approaches instead of knowledge-structure based approaches. While word class distributions standing behind linguistic features normally are not known beforehand, clustering enables to identify new, unknown linguistic features from text. Semantic clustering does not identify word class distributions of texts using similar terms as it is normally done by knowledge-structure based approaches. It identifies word class distributions from text patterns that are semantically related e.g. text patterns with a common meaning. This is in contrast to the related work.

The new approach is described theoretically and it is applied in a case study. Scientific articles are used to ensure homogeneous writing styles of the authors. Further, articles from the same domain are selected to ensure that characteristic key words from different domain do not occur within the articles. Case study results show the feasibility of the proposed approach by comparing it to the frequent baseline.

Despite the success of linguistic feature in classification of the scientific articles in the case study, future research should focus on aspects of combining shallow and linguistic features. Moreover, the results of this research can be used to investigate the evolution of the language e.g. the academic language. A further avenue of research is to consider additional factors such as nationality of the authors, the gender of the authors, and the scientific domain of the articles for the case study.

Future work also can improve evaluation by varying the determined impact threshold for selecting relevant terms. This probably changes the word class distribution for each feature. Further, future word also can vary the threshold for the simple Euclidean distance. This also has an influence on the linguistic features.

ACKNOWLEDGEMENT

We thank Springer publisher who provides us institutional access to its book series and university of Malaya for supporting this research with Research Grant RP026-14AET and PG031-2014A.

REFERENCES

[1] T. Joachims, “Text categorization with Support Vector Machines: Learning with many relevant features,” in *Lecture Notes in Computer Science*, 1998, pp. 137–142.

[2] J. Rybicki, “Burrowing into Translation: Character Idiolects in Henryk Sienkiewicz’s Trilogy and its Two English Translations,” *Lit. Linguist. Comput.*, vol. 21, no. 1, pp. 91–103, Nov. 2005.

[3] M. Koppel, “Automatically Categorizing Written Texts by Author Gender,” *Lit. Linguist. Comput.*, vol. 17, no. 4, pp. 401–412, Nov. 2002.

[4] L. B. Huang, V. Balakrishnan, R.G. Raj, "Improving the relevancy of document search using the multi-term adjacency keyword-order model." *Malaysian Journal of Computer Science*, Vol. 25, No. 1, 2012, pp. 1-10.

[5] A. Qazi, R. G. Raj, M. Tahir, M. Waheed, S. U. R. Khan, and A. Abraham, “A Preliminary Investigation of User Perception and Behavioral Intention for Different Review Types: Customers and Designers Perspective,” The Scientific World Journal, vol. 2014, Article ID 872929, 8 pages, 2014. doi:10.1155/2014/872929.

[6] P. Lv, G.-F. Wang, Y. Wan, J. Liu, Q. Liu, and F. Ma, “Bibliometric trend analysis on global graphene research,” *Scientometrics*, vol. 88, no. 2, pp. 399–419, Aug. 2011.

[7] S. Degaetano-ortlieb, P. Fankhauser, H. Kermes, E. Lapshinova-koltunski, N. Ordan, and E. Teich, “Data Mining with Shallow vs . Linguistic Features to Study Diversification of Scientific Registers,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC ’14)*, 2014, pp. 1327–1334.

[8] C. Whitelaw and J. Patrick, “Selecting systemic features for text classification,” in *Proceedings of the Australasian Language Technology Workshop*, 2004, pp. 93–100.

[9] A. F. Hu, Xiao and Downie, J Stephen and Ehmann, “Lyric text mining in music mood classification,” in *10th International Society for Music Information Retrieval Conference*, 2009, p. 411.

[10] S.-T. Wu, Y. Li, and Y. Xu, “Deploying Approaches for Pattern Refinement in Text Mining,” in *Data Mining, 2006. ICDM ’06. Sixth International Conference on*, 2006, pp. 1157–1161.

[11] J. Kupiec, “Robust part-of-speech tagging using a hidden Markov model,” *Comput. Speech Lang.*, vol. 6, no. 3, pp. 225–242, Jul. 1992.

[12] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, “A Practical Part-of-speech Tagger,” in *Proceedings of the Third Conference on Applied Natural Language Processing*, 1992, pp. 133–140.

[13] A. Ratnaparkhi, “A maximum entropy model for part-of-speech tagging,” in *Proceedings of the conference on empirical methods in natural language processing*, 1996, pp. 133–142.

[14] M. Moohebat, R. G. Raj, S. B. A. Kareem, and D. Thorleuchter, “Identifying ISI-indexed articles by their lexical usage: A text analysis approach,” *J. Assoc. Inf. Sci. Technol.*, vol. 66, no. 3, pp. 501–511, 2015.

[15] S. Bergsma, M. Post, and D. Yarowsky, “Stylometric Analysis of Scientific Articles,” in ’2 Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2012, pp. 327–337.

[16] S. Afroz, M. Brennan, and R. Greenstadt, “Detecting Hoaxes, Frauds, and Deception in Writing Style Online,” in *2012 IEEE Symposium on Security and Privacy*, 2012, pp. 461–475.

[17] J. Jiang, M. W. Berry, J. M. Donato, G. Ostrouchov, and N. W. Grady, “Mining consumer product data via latent semantic indexing,” *Intell. Data Anal.*, vol. 3, no. 5, pp. 377–398, 1999.

[18] Q. Luo, E. Chen, and H. Xiong, “A semantic term weighting scheme for text categorization,” *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12708–12716, Sep. 2011.

[19] H.-H. Tsai, “Global data mining: An empirical study of current trends, future forecasts and technology diffusions,” *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8172–8181, Jul. 2012.

[20] K. Christidis, G. Mentzas, and D. Apostolou, “Using latent topics to enhance search and recommendation in Enterprise Social Software,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9297–9307, Aug. 2012.

[21] T. Hofmann, “Probabilistic latent semantic indexing,” in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’99, 1999, pp. 50–57.

[22] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[23] D. D. Lee and H. S. Seung, “Algorithms for Non-negative Matrix Factorization,” in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 556–562.

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” . *J. Mach. Learn. Res.*, vol. 3, no. 4–5, pp. 993–1022, 2003.

[25] E. H. Ramirez, R. Brena, D. Magatti, and F. Stella, “Topic model validation,” *Neurocomputing*, vol. 76, no. 1, pp. 125–133, Jan. 2012.

[26] G. K. Zipf, *Human behaviour and the principle of least effort*. Cambridge: Addison-Wesley, 1949.

[27] G. Salton, J. Allan, and C. Buckley, “Automatic structuring and retrieval of large text files,” *Commun. ACM*, vol. 37, no. 2, pp. 97–108, Feb. 1994.

[28] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[29] D. Thorleuchter, T. Scheja, and D. Van den Poel, “Semantic weak signal tracing,” *Expert Syst. Appl.*, vol. 41, no. 11, pp. 5009–5016, Sep. 2014.

[30] J. Zhong and X. Li, “Unified collaborative filtering model based on combination of latent features,” *Expert Syst. Appl.*, vol. 37, no. 8, pp. 5666–5672, Aug. 2010.

[31] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, “Indexing by latent semantic analysis,” *J. Assoc. Inf. Sci. Technol.*, vol. 41, no. 6, pp. 391–407, 1990.

[32] V. L. Miguéis, D. Van den Poel, A. S. Camanho, and J. Falcão e Cunha, “Modeling partial customer churn: On the value of first product-category purchase sequences,” *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11250–11256, Sep. 2012.

[33] D. . D. Benoit and D. Van den Poel, “Improving customer retention in financial services using kinship network information,” *Expert Syst. Appl.*, vol. 39, no. 13, pp. 11435–11442, 2012.

[34] K. . Jones, “A statistical interpretation of term specificity and its application in retrieval,” *J. Doc.*, vol. 28, no. 11–21, 1972.

[35] E. J. Halpern, M. Albert, A. M. Krieger, C. E. Metz, and A. D. Maidment, “Comparison of receiver operating characteristic curves on the basis of optimal operating points,” *Acad. Radiol.*, vol. 3, no. 3, pp. 245–253, Mar. 1996.

[36] A. R. van Erkel and P. M. T. Pattynama, “Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology,” *Eur. J. Radiol.*, vol. 27, no. 2, pp. 88–94, May 1998.

[37] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve.,” *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.