Comparing support vector machine and multinomial naive Bayes for named entity classification of South African languages

W. Fourie
Centre for Text Technology
North-West University, Potchefstroom Campus
Potchefstroom, South Africa
wildrich.fourie@nwu.ac.za

J.V. Du Toit & D.P. Snyman
School for Computer, Statistical and Mathematical Sciences
North-West University, Potchefstroom Campus
Potchefstroom, South Africa
{tiny.dutoit; dirk.snyman}@nwu.ac.za

***Abstract*—In this study, two classical machine learning algorithms, multinomial naive Bayes and support vector machines, are compared when applied to named entity recognition for two South African languages, Afrikaans and English.**

The definition of a named entity was based on previous definitions and deliberations in literature as well as the intended purpose of classifying sensitive personal information in textual data. For the purpose of this study, the best algorithm should be able to deliver accurate results while requiring the least amount of time to train the classification model. A binary nominal class was selected for the classifiers and the standard implementation of the algorithms were utilised; no parameter optimisation was done.

All the models achieved remarkable results in both ten-fold cross validation and independent evaluations with the support vector machine models outperforming the multinomial naive Bayes models. The multinomial naive Bayes models, however, required less time to train and would be more suited to low resource implementations.

Keywords—binary class; cross-domain; named entity classification; multilingual; multinomial naive Bayes; support vector machines

I. INTRODUCTION

Digital textual data resources for South African languages are very rare compared to other available international corpuses [[1], [2], [3]]. In a bid to address this issue, the South African Government's Department of Arts and Culture (DAC) funded and launched the National Centre for Human Language Technologies' (NCHLT) Resource Management Agency (RMA; [4], [5]). The centre is based on similar centres internationally and provides a sustainable step towards providing resources for research and development in Human Language Technology (HLT). The aim of the centre is to provide a centralised platform for the distribution of Natural Language Processing (NLP) resources such as text and audio corpora [5]. One problem faced by such centres is the anonymisation of private information contained in data sourced from private companies, organisations and publishing houses.

During anonymisation, private and personal information such as telephone numbers, addresses (residential, postal, e-mail), values of currency and named entities (NEs) are removed or replaced with predefined or generated information. This is done to protect the individual or organisation from attempts to derive the information by examining the publicly published corpus. While the numbers and addresses are easily identified using regular expressions and lists, the classification of NEs is a more imposing problem concerning the plethora of organisations, names, surnames and other subjective entities such as president, colonel, health ministry, Autshumato project, Mona Lisa and Jurassic period for example.

Computerised learning techniques have shown acceptable to remarkable results in NE classification [[6], [7], [8], [9], [10], [11]], although Nadeau and Sekine [12] argue that comparisons between results are difficult due to differences in evaluation techniques. This study seeks to report the results of applying two specific classification algorithms for NE classification, with the aim of anonymisation, on two South African languages. The article is organised as follows: a brief overview of similar investigations is given in Section II, followed by the experimental setup in Section III. Results from the experiments are presented in Section IV and finally in Section V, some conclusions are drawn.

II. RELATED WORK

Information extraction (IE) is the extraction of useful information from raw data sets in order to aid in decision-making and the automation of certain processes [13]. This varied field includes disciplines such as image recognition, text classification, biomedical classification and data mining. This study will focus on one specific branch of text classification known as named entity recognition and classification (NERC).

The aim of NERC systems is the recognising and classifying of predefined textual units which are referred to as NEs [[7], [14], [15], [16], [17]]. The identified units are classified using predefined classes of NEs and subsequent mark up for each. The sentence

"Mr. Kroon, from GlobalCorp, can be contacted directly at 012 555 5555."

can be classified as

<PER>Mr. Kroon</PER>, from <ORG>GlobalCorp</ORG> can be contacted directly at <NUM>012 555 5555</NUM>.

The <PER></PER> tag set indicates a person, <ORG></ORG> an organisation and <NUM></NUM> a number. For removing confidential information from texts, classified units can be replaced by blank or randomised values from the same class (person, organisation and number).

The term NE was first defined by the sixth Message Understanding Conference (MUC) in 1995 [[7], [12], [18], [19]] and expanded for the seventh MUC [21]. The aim of the NE shared tasks of the MUC-6 and MUC-7 conferences in 1995 and 1997 respectively, was to assign several teams with the NERC of supplied data sets. For the tasks, a structured definition of an NE was provided together with training and testing data as well as evaluation metrics [[12], [19], [21], [22], [23]]. Marrero et al. [7] note that most current NERC systems are built on the basis for NEs as laid out by the MUC shared tasks.

Puttkammer [10] details the only attempt at NERC for a South African language (Afrikaans), aided by the use of gazetteers [[11], [18]]. His hybrid system achieved an F1-measure of 0.9474. The survey of NERC [12] is recommended for further reading into the history and scope of NERC research. In [7] a recent and excellent overview of NERC research is provided. In addition, key faults of previous investigations are discussed.

Next, the experimental setup is explained by detailing the definition of an NE, the algorithm selection, corpora used, experimental toolkit and configuration as well as the evaluation criteria.

III. Experimental setup

A. Definition of a named entity

The MUC defines the NE task as follows: "The Named Entity task consists of three subtasks (entity names, temporal expressions, number expressions). The expressions to be annotated are 'unique identifiers' of entities (organisations, persons, locations), times (dates, times), and quantities (monetary values, percentages)" [[21], [22], [23]]. A set of words and numbers representing a duration or point in time is defined as a temporal expression. Alhelbawy and Gaizauskas' [24] definition as well as that of Puttkammer [10] was based closely on the MUC definition. Although the MUC shared tasks delivered a reused basis for the definition of an NE, multiple versions and deviations exist in previous work. Borrega et al. [19] attribute the varied differences in NE definitions to the separate restrictions required to implement the NERC system practically. The evolution of the definition to suit the domain and purpose is evident in the literature and is, according to Marrero et al. [7], "the only one constant" in the aim to define an NE.

As with the MUCs, the definition is based on its intended purpose; additions to the definition are based on the examination of the corpus. With the aim of identifying sensitive information this study defines NEs as phrases that contain the names of persons, organisations, locations, time and quantities [[20], [23]]. It includes official status (president, general, colonel), non-profit organisation (NPO) names, laws, acts, product names, public holidays, seasons, scientific measures, titles, government departments and forms, educational institutions and courses, language names, past or ongoing project names, denominators and values of currency, dates in written and decimal form, telephone numbers, ID numbers, any addresses (e-mail, website, residential, business, home), and quantities. General knowledge terms or information that was readily available was not included in the NE definition. The following entities did not reveal specific information in this domain: names of plants, animal and bug species, scientific names; and general directions (north, east, south, west).

A single NE will constitute the longest possible sequence of words that can be viewed as a single entity. For example, the sequence: "14 Boom Street, Klerksdorp, South Africa", is recognised as a single NE since it describes a single entity. Although most temporal expressions could be sufficiently handled practically using language-specific regular expressions [19] and gazetteers [[11], [18]] the combination of these techniques with an automated classification system could improve the accuracy of an anonymisation system. This definition forms a basis for the intended purpose of building a working NERC system to annotate textual resources in the English and Afrikaans languages.

Next, the selection of classification algorithms is discussed.

B. Algorithm selection

The support vector machine (SVM) is considered the most accurate general-purpose classifier for pattern recognition, but can be computationally expensive when faced with very large data sets [25]. This technique was first proposed by Vapnik [26] and conceptualised by Vapnik and Cortes [27]. SVMs do not rely on probabilities to build a classification system. Instead, binary class assignment is used, which represents data points as high but finite dimensional vectors [[26], [27], [28]]. For the p-dimensional vectors an optimal (p-1)-dimensional hyperplane is sought, one which maximises the distance or margin between the different classes [[25], [26], [27]]. The vectors that best define the separation of the classes are designated as the support vectors and the optimal separating hyperplane function is defined by these support vectors. Slack variables as well as the kernel trick are applied when the data cannot be separated linearly [[25], [28]]. The SVM algorithm has been selected since classification only makes use of the limited number of support vectors identified during the training of the system. A small corpus might be enough to build a functional and competing classification system. The ability of SVMs to generalise easily might make them adaptable between domains and languages.

Zhang [8] states that the naive Bayes (NB) type of Bayesian network has delivered "surprisingly good classification performance", a belief supported by McCallum and Nigam [29]. Traditionally, two first-order probabilistic naive Bayes assumption-based models are used: the multivariate Bernoulli

model and the multinomial naive Bayes (MNB) model [29]. The multivariate Bernoulli model is based on the occurrence of a text unit in a textual resource (document, paragraph, sentence); the frequency and order of occurrences are not considered, only whether a text unit is present or not. The multinomial model is also not concerned with the order of the text units in the resource, but it does include the frequencies of occurrences. McCallum and Nigam [29] have demonstrated that the multivariate Bernoulli model fares better for small vocabularies but is outperformed by the multinomial model before a vocabulary of 1000 words is reached. The multinomial model also fares better with classifying text units that vary in length. A formal definition of the naive Bayes probability equation for NERC purposes is given in [30]. The probability that the current inspected "word" (or sequence of words) is an NE, is equal to the probability of all NEs in the text multiplied by the product of the probabilities of each word in the text being an NE. Similar to SVMs, MNB algorithms have shown remarkable results using small corpora for classification. The MNB algorithm however is not as computationally complex as the SVM algorithm and is well suited to practical implementations.

The MNB and SVM machine learning algorithms have been shown to deliver reasonably acceptable classification results while using minimal textual resources [[2], [6], [8], [29]].

In the next section, the corpora and its attributes are discussed.

C. Corpora

Two separate data sets were obtained; the first a parallel corpus of Afrikaans and English texts and the second an annotated Afrikaans word corpus. The official ISO 639-3 language code [31] and ISO 3166-1 country code [32] combinations for South African English (ENG-GB) and Afrikaans (AFR-ZA) were used. The first corpus was provided in 233 separate AFR-ZA and ENG-GB documents, aligned on sentence level. The second corpus was provided in a single comma separated values (CSV) document. Each line contained a word and Boolean value; the words followed chronologically from the original government domain texts.

The first corpus originated from a local magazine which publishes in several languages. All of the separate documents for each language were merged, in parallel, into two sentence-aligned documents. Automatic annotation methods were first used to retrieve an initial gazetteer from the given text. The automatically annotated texts and gazetteers were then checked by a native language speaker in either language. The languages were not checked in parallel although several similarities existed such as person names, numbers, business and location names. The revised gazetteers were then used to classify the NEs in the original texts. This bootstrapping process was repeated iteratively until all noticeable and discernible NEs were classified. The annotated corpus is assumed not to be of a Gold Standard [33].

The Stanford Named Entity Recogniser (SNER, [34]) and Autshumato Text Anonymiser (ATA, [35]) were used to automatically annotate the corpus before the first iteration. The SNER annotation used the supplied 7-class MUC, 4-class CoNLL (defined by the Conference on Natural Language Learning [20]) and 3-class combined models. The flexible nature of the ATA application allowed the inclusion of language specific (and non-specific) lists and rules for classification. Currently the ATA application does not utilise any machine learning model in the classification process; it relies on user-supplied gazetteers and customisation of the rules. The data was annotated incrementally with each entity not recognised in a step, being included in the custom lists for the next annotation iteration. Finally, the automatically annotated sentence-level English and Afrikaans documents were checked manually, and any entities falsely classified or not classified, were corrected.

At this stage it was noted that the number of classified NEs differed between the Afrikaans and English texts. This could be attributed to error during translation or annotation. The longest combination of words that represented an NE for both English and Afrikaans was seven words. The annotated data was then processed by splitting the texts into word n-gram windows between 3-grams and 7-grams and outputting separate documents for each language and n-gram. Three-gram windows were chosen as the lowest granularity since they can already be considered too small to sufficiently include the context around a word [14]. Up to 7-gram granularity was chosen since the longest single NE found in the data consisted of seven words. Additionally, word-separated and sentence-separated documents were created. Duplicates were not removed from any of the data sets so as not to distort the occurrence frequency which should aid in disambiguation. Table III indicates the number of instances per language for each class: textual units containing NEs and units not containing NEs. The number of instances for each language was different across all of the different levels of granularity. It indicates that for this set of data all of the NEs were not directly mapped across the languages, although many similarities did occur.

Next the experiment toolkit and implementations of the algorithms are discussed.

D. The WEKA toolkit

The WEKA toolkit [[30], [36], [37]] was used in conducting the experiments using the supplied implementations of the MNB and SVM algorithms. The WEKA implementation of an SVM classifier is applied through Platt's Sequential Minimal Optimisation (SMO) algorithm [36], which breaks up the large complex quadratic programming (QP) problem posed by SVM training [27] into smaller, more easily computable QP problems [38].

The data was converted, with the aid of the WEKA toolkit, with a string-to-word vector filter. Each of the words found in the data is defined as a class. The strings are converted to decimal arrays; a single decimal value in the array maps to a class defined word. The word classes were not lowercased or balanced. In balanced classes, the frequency of occurrence has been removed so as not to skew the model towards one particular class per instance. As several words are shared among NEs and non-NEs, an unbalanced approach is required

for accurate classification. For example, consider the following sentence: “Mr. Ward was allowed to visit the children’s ward.” A person might have the surname Ward, which also refers to a specific room in a hospital. By removing separate occurrences of the word “ward”, the instance of the word as a surname would also be removed, resulting in the misclassification of Mr. Ward instances.

The techniques and metrics used to evaluate the algorithms are discussed next.

E. Evaluation

The converted data sets for each combination of language and granularity were fed to the WEKA toolkit and trained on both the SVM and MNB algorithms using the default parameters. The results were evaluated using a stratified ten-fold cross-validation test producing confusion matrixes for each set. An explanation of a confusion matrix is given in Table I. True positive (TP) is the sum of units (n-gram, sentence, word) containing one or several NEs which were classified as containing NEs. False Positive (FP) is the number of units containing NEs that were not classified as containing NEs. False Negative (FN) is the sum of units that do not contain NEs but were classified as containing NEs and True Negative (TN) is the number of non-containing NE units classified correctly.

TABLE I. CONFUSION MATRIX

Model \ Actual classification	**Contains NE(s)**	**Does not contain NE(s)**
Contains NE(s)	True Positive (TP)	False Positive (FP)
Does not contain NE(s)	False Negative (FN)	True Negative (TN)

The WEKA toolkit represents classification, n-fold cross validation and independent test set evaluations in industry accepted precision, recall and F1-measures [[1], [2], [6], [10], [14], [39]] as originally defined by [40]. The formulas for recall and precision are given in (1) and (2). The F1-measure (3) provides a weighted harmonic mean between the recall and precision; an equal weight assignment is used in this study. In the case of n-fold cross validation, the results of each iteration is averaged into the final results [36].

$$\text{Recall (R)} = TP / (TP + FN); \quad (1)$$

$$\text{Precision (P)} = TP / (TP + FP); \text{ and} \quad (2)$$

$$\text{F1-measure} = 2(R \times P) / (R + P). \quad (3)$$

A statistical significance comparison was done on the 3-gram MNB and SVM model for the AFR-ZA corpus utilising the Experimenter from the WEKA-toolkit. The modified T-Test evaluation method used is referred to by Bouckaert et al. [36] as the “corrected resampled T-Test”.

IV. RESULTS

The time taken to train each model is given in Table II, these values are not explicitly accurate as various other background processes can influence the time required to train the model. Problems with this measure are evident and oppose the number of entities contained within the data set, shown in Table III. The 3-gram, 4-gram and 5-gram AFR-ZA data set contained a total of 63254, 59283 and 55490 entities respectively, while showing fluctuations in MNB times and increases in the SVM times. For the ENG-GB data, the MNB models recorded similar results to the AFR-ZA data whereas the SVM models showed some drastic differences in the 6-gram and 7-gram data. Although the fluctuation of times clearly indicates noticeable inaccuracy in the “time to train” measure, noticeable differences can be spotted across the investigated algorithms. This is good enough to draw broad conclusions on the time required to train a specific data set.

The results of the language/granularity evaluation are shown in Table IV. The accuracy of the MNB and SVM models decreased as the granularity level increased. The biggest decline was noticed in the Afrikaans models; the SVM declined from a 0.994 to 0.992 F1-measure and the MNB from a 0.988 to 0.978 F1-measure, a difference of 0.002 and 0.010 respectively. Across the granularity levels, the worst (although adequate) results were obtained from the word-level and sentence-level models, again for Afrikaans. The SVM word model achieved an F1-measure of 0.909 and the sentence model an F1-measure of 0.923; with 0.904 and 0.913 for MNB words and sentences. The SVM models fared better than the MNB models across all of the granularity levels, although the differences seem marginal. The SVM models required exhaustive computational resources and time to complete the training of the model whereas the MNB algorithm delivered excellent results using minimal time to build the models.

In Afrikaans, the SVM models outperformed the MNB models and the best SVM models were the 3-gram to 5-gram models each obtaining an F1-measure of 0.994, which is quite remarkable. The best results for the English data are the 3-gram to 6-gram SVM models, each with an F1-measure of 0.995, which is 0.6% better than the best MNB result, the 3-gram model. The results of both the MNB and SVM models are almost mirrored for both languages – which might be an indication of the similarity of NEs found between the two languages.

Taking all of the previous results into consideration, the 3-gram models are deemed the most accurate for delivering the best or equal to the best F1-measure as well as requiring the least amount of training time for the SVM algorithm. Although the 3-gram models have more class instances than other n-gram models (Table III), the instances are shorter and less expensive to convert and train. Based on these deliberations, the 3-gram models are chosen as the most accurate and are used for the independent test. The word-level models are also included since they delivered adequate results, using the least amount of time to train, and being able to deliver a practical classifier.

The results from the granularity/language test are suspiciously high: 99.5% for the best SVM model and 98.8% for the best MNB model. To verify the accuracy of the results,

an independent test was conducted; the trained MNB and SVM models for the 3-gram and word level AFR-ZA models were used and evaluated on the annotated, government domain corpus. The results of the experiments for each of the training algorithms and data sets are also given in Table IV.

The MNB narrowly outperformed the SVM model and achieved an F1-measure of 0.894 as opposed to 0.893 for the 3-gram model. This model could efficiently be applied across the two separate domains of Afrikaans. The speed at which the model could be trained also enables the use of this machine learning algorithm in instantly re-trainable systems. It should be noted that the word model also delivered surprisingly good results, indicating that although the use of gazetteers can greatly speed up the annotation process and aid in classification, their explicit use is not required. A model trained on data annotated by the means of gazetteers was able to accurately identify NEs in another data set without the use of the annotation gazetteers.

The results from the statistical significance test between the AFR-ZA 3-gram MNB and SVM models indicated that the SVM model is statistically significant when compared to the MNB model.

V. Conclusion

This study aimed to compare two statistical machine learning algorithms at the task of identifying NEs contained in textual resources for two South African languages, English and Afrikaans. A binary nominal class was selected and the best model should only be able to determine if an investigated textual unit is an NE or not. The algorithm must be expandable to other domains, and not depend on language-specific linguistic rules and definitions. The definition of an NE was based on previous relevant definitions and expanded to include occurrences in the domain-specific data.

Owing to the scarceness of aligned multilingual data for South African languages, the choice of the domain was necessitated by the availability of the data. A parallel aligned English-Afrikaans magazine article corpus was obtained, as well as an annotated Afrikaans corpus in the government domain. The parallel corpus, originally in separate parallel documents, was converted and annotated using an iterative, bootstrap technique. Several data sets were produced from this corpus to evaluate the best granularity to use when classifying unknown text segments.

The choice of algorithms was based on their ability to suit the restrictions in training data as well as previously reported results. The SVM only slightly outperformed the MNB models across all granularity levels and both languages. Because it is computationally expensive, this model would be suited in instances where the NERC system uses a fixed, pre-trained model. The MNB models delivered results as high as the SVM models with less time required to train the models. The word and sentence models achieved reasonable results, and MNB word and sentence models could easily be implemented as low-resource, re-trainable, early NE detectors that could quickly scan an incoming text. More accurate and expensive NERC systems could then be launched if an NE was detected. The information contained within the grammatical structure is well maintained with the n-gram models, delivering higher results. For the practical application of anonymisation of private information in textual resources, an MNB re-trainable 3-gram model, with the assistance of gazetteers, will be used. The MNB models deliver excellent results and use far less resources than their relevant SVM models, which allow them to be easily retrained on recently classified data.

This study was limited to focusing on two similar South African languages. Studies of other related languages would more clearly indicate cross-lingual adaptability of the algorithms. The specific NE definition required for any study limits its comparability to other similar systems and is reported to limit these models to certain domains. Although a host of other multilingual NERC approaches exist, remarkable results can be obtained by good definition, adequate corpus and classical classification algorithms such as SVM and MNB. It would also be interesting to extend this study to include more South African languages, especially sharing similarities between them. The development of NERC systems for all of the South African languages could assist in building useful annotated corpora for natural language processing and human language technology research.

Acknowledgment

We wish to express our gratitude to Dr. Martin Puttkammer and Dr. Roald Eiselen for their expert advice as well as to the Centre for Text Technology (CTexT®) for providing the data.

References

[1] D.P. Snyman, G.B. Van Huyssteen and W. Daelemans, "Cross-Lingual Genre Classification for Closely Related Languages," in Proc. PRASA, 2012, pp. 133-137.

[2] D.P. Snyman, G.B. Van Huyssteen and W. Daelemans, "Automatic Genre Classification for Resource Scarce Languages," in Proc. PRASA, 2011, pp. 132-137.

[3] A. Grover, G.B. Van Huyssteen and M. Pretorius, "The South African human language technology audit," Language Resources and Evaluation, vol. 45, no. 3, 2011, pp. 271-288.

[4] CTexT (Centre for Text Technology). (2012). Resource Management Agency Newsletter 1 of 2012 [Online]. Available: http://rma.nwu.ac.za/images/stories/pdfs/News.RMA.Newsletter.1.0.1.MHM.2012-12-11.pdf

[5] M. Muller. (2012, March 26). Good news for South African languages [Online]. Available: http://www.researchsa.co.za/news.php?id=1053

[6] N. Jahan and S. Morwal, "Named Entity Recognition in Indian languages: a sruvey," Int. J. Engineering Sciences and Research Technology, vol. 2, no. 4, 2013, pp. 925-929.

[7] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato and J.M. Gómez-Berbís, "Named Entity Recognition: Fallacies, challenges and opportunities," Computer Standards & Interfaces, vol. 35, no. 5, 2013, pp. 482-489.

[8] H. Zhang, "The optimality of naive Bayes," in Proc. 7th Int. Florida Artificial Intelligence Research Society (FLAIRS) Conf., AAAI, 2004, pp. 3-9.

[9] X. Ma, "Toward a name entity aligned bilingual corpus," in Proc. LREC, 2010, pp. 17-23.

[10] M.J. Puttkammer, "Automatic Afrikaans tokenisation," M.A. dissertation, School of Languages, North-West Univ., Potchefstroom, South-Africa, 2006.

[11] A. Mikheev, M. Moens and C. Grover, "Named entity recognition without gazetteers," in Proc. 9th Conf. European chapter of the Association for Computational Linguistics, ACL, 1999, pp. 1-8.

[12] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” Lingvisticae Investigationes, vol. 30, no. 1, 2007, pp. 3-26.

[13] D. Jurafsky and J.H. Martin, Speech & language processing: an introduction to natural language processing, computational linguistics, and speech recognition, Prentice Hall, 2000.

[14] R. Al-Rfou and S. Skiena, “SpeedRead: A Fast Named Entity Recognition Pipeline,” arXiv preprint arXiv:1301.2857, 2013.

[15] H.N. Goh, L.K. Soon and S.C. Haw, “Automatic identification of protagonist in fairy tales using verb,” Advances in Knowledge Discovery and Data Mining, P. Tan, S. Chawla, C.K. Ho and J. Baily eds., Springer Berlin, 2012, pp. 395-406.

[16] M. Marcińczuk and M. Janicki, “Optimizing CRF-based model for proper name recognition in Polish texts,” in Proc. Computational Linguistics and Intelligent Text Processing, Springer, 2012, pp. 258-269.

[17] D.M. Nemeskey and E. Simon, “Automatically generated NE tagged corpora for English and Hungarian,” in Proc. 4th Named Entity Workshop, Association for Computational Linguistics (ACL), 2012, pp. 38-46.

[18] J. Nothman, N. Ringland, W. Radford, T. Murphy and J.R. Curran, “Learning multilingual named entity recognition from Wikipedia,” Artificial Intelligence, vol. 194, 2013, pp. 151-175.

[19] O. Borrega, M. Taulé and M.A. Marti, “What do we mean when we speak about Named Entities,” in Proc. Corpus Linguistics Conference, 2007.

[20] E.F. Tjong Kim Sang and F. De Meulder, “Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition,” in Proc. 7th Conf. on Natural Language Learning at HLT-NAACL 2003, Association for Computational Linguistics, pp. 142-147.

[21] N. Chinchor and P. Robinson, “MUC-7 named entity task definition,” in Proc. 7th Conference on Message Understanding (MUC-7), 1997.

[22] R. Grishman and B. Sundheim, “Message Understanding Conference-6: A Brief History,” in Proc. COLING, Morgan Kaufman, 1996, pp. 466-471.

[23] R. Grishman and B. Sundheim. (1995, March 21). Sixth Message Understanding Conference (MUC-6): conference task definition [Online]. Available: http://www.cs.nyu.edu/cs/faculty/grishman/COtask21.book_1.html

[24] A. Alhelbawy and R. Gaizauskas, “Named entity based document similarity with svm-based re-ranking for entity linking,” in Proc. Advanced Machine Learning Technologies and Applications, Springer, 2012, pp. 379-388.

[25] C.J. Van Heerden, “Efficient training of support vector machines and their hyperparameters,” Ph.D. dissertation, School of Electrical, Electronic and Computer Engineering, Nort-West Univ., Potchefstroom, South-Afirca, 2012.

[26] V.N. Vapnik, B.E. Boser and I.M. Guyon, “. A training algorithm for optimal margin classifiers,” in Proc. 5th Annu. Workshop on Computational Learning Theory, ACM, 1992, pp. 144-152.

[27] C. Cortes and V.N. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, 1995, pp. 273-297.

[28] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical Recipes: The art of scientific computing, 3rd ed. New York: Cambridge University Press, 2007, pp. 883-898.

[29] A. McCallum and K. Nigam, “A comparison of event models for naive bayes text classification,” in Proc. AAAI-98 workshop on learning for text categorization, Madison, WI: Citeseer, 1998, vol. 752, pp. 41-48.

[30] W. Ertel, Introduction to artificial intelligence, N. Black ed., London, UK: Springer, 2011, pp. 202-206.

[31] Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages, ISO 639-3, 5 February, 2007.

[32] Codes for the representation of names of countries and their subdivisions, ISO 3166-1 alpha-2, 1974.

[33] L. Wissler, M. Almashraee, D. Monett and A. Paschke, “The Gold Standard in Corpus Annotation,” in Proc IEEE Germany Student Conference 2014 [Online]. Available: http://www.ieee-student-conference.de/fileadmin/templateConf2014/images/papers/ieeegsc2014_submission_3.pdf

[34] The Stanford Natural Language Processing Group. Stanford Named Entity Recognizer (NER), ver. 1.2.8. Stanford, CA: Stanford University, 2013.

[35] CTexT. Autshumato Text Anonymiser (ATA), ver. 2.0.0. Potchefstroom: Nort-West University, 2012.

[36] R.R. Bouckaert, E. Frank, M. Hall, P. Kirby, P. Reutemann, A. Seewald and D. Seuse. (2013). WEKA Manual for Version 3-7-10 [Online]. Available: http://ufpr.dl.sourceforge.net/project/weka/documentation/3.7.x/WekaManual-3-7-10.pdf

[37] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten, “The WEKA data mining software: an update,” ACM SIGKDD explorations newsletter, vol. 11, no. 1, 2009, pp. 10-18.

[38] J.C. Platt, “Fast training of support vector machines using sequential minimal optimization,” Advances in kernel methods, B. Schoelkopf, C. Burges and A. Smola eds., MIT press, 1999, pp. 185-208.

[39] N. Kang, E.M. Van Mulligen and J.A. Kors, “Training text chunkers on a silver standard corpus: can silver replace gold?,” BMC bioinformatics, vol. 13, no. 1, 2012, pp. 17-22.

[40] C. Van Rijsbergen, Information retrieval, 2nd ed. London, UK: Butterworth-Heinemann, 1979.

TABLE II. TIME TAKEN TO TRAIN EACH MODEL

Dataset	Time (seconds)	
AFR-ZA	***MNB***	***SVM***
3-gram	0.08	1200.34
4-gram	0.06	1367.74
5-gram	0.08	1445.65
6-gram	0.08	1348.62
7-gram	0.08	622.28
Words	0.00	90.17
Sentences	0.03	29.36
ENG-GB	***MNB***	***SVM***
3-gram	0.06	534.54
4-gram	0.08	698.04
5-gram	0.06	568.08
6-gram	0.06	952.43
7-gram	0.08	824.39
Words	0.00	77.48
Sentences	0.02	23.62

TABLE III. NUMBER OF INSTANCES PER LANGUAGE FOR EACH CLASS

	Language	AFR-ZA		Total	ENG-GB		Total	AFR-ZA independent test		Total
		NE	***Not NE***		***NE***	***Not NE***		***NE***	***Not NE***	
Granularity	***3-gram***	4032	59222	63254	4133	61400	65533	4834	50620	55454
	4-gram	4574	54709	59283	4683	56841	61524	-	-	-
	5-gram	4906	50584	55490	5033	52635	57668	-	-	-
	6-gram	5142	46748	51890	5278	48726	54004	-	-	-
	7-gram	5284	43157	48441	5459	45034	50493	-	-	-
	Word	521	7204	7725	469	6346	6815	2460	52997	55457
	Sentences	985	2925	3910	923	2890	3813	-	-	-

TABLE IV. RESULTS FOR THE NAMED ENTITY RECOGNITION OF TWO LANGUAGES AND DIFFERENT GRANULARITIES

		MNB			SVM		
Language	**Dataset**	**Precision**	**Recall**	**F1-measure**	**Precision**	**Recall**	**F1-measure**
AFR-ZA							
	3-gram	**0.988**	**0.988**	**0.988**	**0.994**	**0.994**	**0.994**
	4-gram	0.986	0.985	0.985	0.994	0.994	0.994
	5-gram	0.983	0.983	0.983	0.994	0.994	0.994
	6-gram	0.983	0.983	0.983	0.993	0.993	0.993
	7-gram	0.978	0.977	0.978	0.992	0.992	0.992
	Words	0.934	0.934	0.904	0.938	0.936	0.909
	Sentences	0.912	0.914	0.913	0.927	0.926	0.923
ENG-GB							
	3-gram	**0.988**	**0.988**	**0.988**	**0.995**	**0.995**	**0.995**
	4-gram	0.986	0.986	0.986	0.995	0.995	0.995
	5-gram	0.984	0.983	0.983	0.995	0.995	0.995
	6-gram	0.981	0.980	0.981	0.995	0.995	0.995
	7-gram	0.979	0.978	0.978	0.994	0.994	0.994
	Words	0.933	0.933	0.903	0.939	0.936	0.910
	Sentences	0.922	0.923	0.922	0.930	0.930	0.928
AFR-ZA independent test							
	3-gram	**0.897**	**0.918**	**0.894**	**0.898**	**0.918**	**0.893**
	Words	0.923	0.956	0.934	0.946	0.958	0.946