Effective and Efficient Classification of Topically-Enriched Domain-Specific Text Snippets:
The TETSC Method

Marco Spruit, Department of Information and Computer Sciences, Utrecht University, Utrecht, The Netherlands

Bas Vlug, Department of Information and Computer Sciences, Utrecht University, Utrecht, The Netherlands

ABSTRACT

Due to the explosive growth in the amount of text snippets over the past few years and their sparsity of text, organizations are unable to effectively and efficiently classify them, missing out on business opportunities. This paper presents TETSC: the Topically-Enriched Text Snippet Classification method. TETSC aims to solve the classification problem for text snippets in any domain. TETSC recognizes that there are different types of text snippets and, therefore, allows for stop word removal, named-entity recognition, and topical enrichment for the different types of text snippets. TETSC has been implemented in the production systems of a personal finance organization, which resulted in a classification error reduction of over 21%. Highlights: The authors create the TETSC method for classifying topically-enriched text snippets; the authors differentiate between different types of text snippets; the authors show a successful application of Named-Entity Recognition to text snippets; using multiple enrichment strategies appears to reduce effectivity.

Keywords: *Classification, Dimensionality Reduction, Enrichment, External Data Sources, Text Snippets, Topic Models*

1. INTRODUCTION: THE WICKED PROBLEM OF CLASSIFYING TEXT SNIPPETS

The recent years have witnessed an unprecedented growth in the amount of text snippets. The Washington Post reports that in March 2013 over 400 million tweets are sent per day, an increase from 200 million since 2011 (Tsukayama, 2013; Twitter Engineering, 2011). This is only the increase in the number of text snippets from one source. In today's society there are plenty of

DOI: 10.4018/IJSDS.2015070101

places where text snippets are found. Twitter is one place mentioned earlier, but also for instance search engines or banks produce a large amount of text snippets per day in the form of search result snippets or financial transactions.

Most of these text snippets are taken as being of no domain. This, however, is far from the truth. There are plenty of domain-related tweets being sent on a daily basis, customer service tweets of companies being one example thereof. Furthermore, there even exist domain-specific search engines, such as MEDLINE, which are designed to yield better results as they are aimed at specific domains.

While a lot of text snippets are created and generated on a daily basis, it currently still is a problem to even only summarize these so-called text snippets through classification. While the classification of large documents has reached effectiveness levels comparable to those of trained professionals, the classification of short texts, in this research denoted as text snippets, is different (Sebastiani, 2002). Chen, Xiaoming & Shen (2011) identify the reason being mainly that text snippets are of short length and therefore suffer from sparsity.

By not being able to correctly classify text snippets, companies miss out on business opportunities. Being able to correctly classify tweets, for instance, could provide a lot of information that can be used to identify trends, or, being able to correctly classify financial transactions could provide account owners with valuable overviews of expenses, which in turn can make them more in control of their finances. Another application domain which is well known to suffer from valuable information in unstructured text snippets, is healthcare, where doctors often record a patient's diagnosis and/or prognosis in the dossier's comment field only (Spruit, Vroon & Batenburg, 2014).

This paper attempts to solve the problem of correctly classifying domain-specific text snippets to predefined categories. A vast amount of literature can be found intended to solve this problem. Most of this literature is related to the enrichment of text snippets through various means:

1. Search query results (*e.g.* Sahami & Heilman, 2006; Shen *et al.*, 2006);
2. The categorical structure of an intermediary (such as Wikipedia or Yahoo, see, *e.g.* Shen *et al.*, 2006; Gabrilovich & Markovitch, 2005);
3. An external corpus (*e.g.* Gabrilovich & Markovitch, 2006; Wang & Domeniconi, 2008);
4. Topic models (*e.g.* Phan, Nguyen & Horiguchi, 2008; Ramage, Dumais & Liebling, 2010); or
5. Lexical information (*e.g.* Hu *et al.*, 2009).

2. TETSC: THE TOPICALLY-ENRICHED TEXT SNIPPET CLASSIFICATION METHOD

In this paper we attempt to solve the problem of correctly classifying domain-specific text snippets to predefined categories through creating TETSC, the Topically-Enriched Text Snippet Classification method. This method is created using the meta-modeling technique of Process-Deliverable Diagrams (PDDs; Weerd & Brinkkemper, 2008). TETSC is applicable in any domain, for example in combination with the Linguistic Engineering for Business Intelligence (LEBI) framework (Otten & Spruit, 2011). TETSC facilitates the usage of any combination of the five aforementioned techniques to text snippet enrichment. Because of this TETSC is a flexible method. To keep the dimensionality increase in the method due to this enrichment to a minimum, the usage of topic models on the enrichment of text snippets is enforced.

Note that we do not introduce new text mining algorithms but rather focus on selecting the most appropriate proven techniques for the task at hand. Vleugel, Spruit & Daal (2010) first proposed this meta-modelling approach to prescribe 'recipes' for decision analytics applications. Pachidi, Spruit & Weerd (2014) first refer to this research approach as the emerging discipline of *meta-algorithmics* (*e.g.* Simske 2013).

Because topic inference on enrichments is enforced, TETSC seems similar to the method created in Phan *et al.* (2008) and Nguyen, Phan & Horiguchi (2009). There is, however, a fundamental difference. In Phan *et al.* (2008) and Nguyen *et al.* (2009) text snippets are enriched through using the topic distribution of the text snippets themselves, inferred on the topic model of a universal dataset. In this paper the intent is to include the topic distribution of an external enrichment based on a topic model created of the dataset this enrichment is from.

Aside from using topic modeling to reduce the dimensionality increase caused by enrichment, TETSC employs stop word removal and Named-Entity Recognition (NER) to reduce the overall dimensionality. Hereby stop words are words that do not convey any significant semantics to the texts or phrases they appear in, and are removed in most studies, such as Hu *et al.* (2009), Dragut *et al.* (2009), and Carpineto & Romano (2012). NER is defined as a form of information extraction in which we seek to classify every word in a document as being a person name, organization, location, date, time, monetary value, percentage, or "none of the above" (Borthwick, 1999), and, although performing poorly when applied to tweets (*e.g.* Ritter, Clark & Etzioni, 2011), is included as multiple machine learning techniques as well as handcrafted rule-based algorithms towards NER exist. Ritter *et al.* (2011) use only one implementation towards NER—the Stanford NER—leaving plenty alternatives untested.

2.1. Process-Deliverable Diagram

Figure 1 shows the Process-Deliverable Diagram (PDD) of TETSC. The corresponding concepts and activities Tables are found in Appendix A and B respectively. TETSC starts off with an initial TEXT SNIPPET CORPUS. It is implied that this TEXT SNIPPET CORPUS consists of a reasonable amount of TEXT SNIPPETS that have already been categorized. This is needed to train the classifiers. If no historical data is available, a subset of the to-be-categorized dataset can be selected and manually categorized. It is important that no category is left empty and that a reasonable training sample per category is used to determine the viability of the employed techniques. In CLASSIFIER training empty categories are usually discarded. Shen *et al.* (2006), for instance, removes every category with less than 10 TEXT SNIPPETS training data, as otherwise no viable classification to these categories can be made.

In TETSC first the domain of the TEXT SNIPPET CORPUS is to be determined, as well as the categories TEXT SNIPPETS can be classified to. Next, it is key to identify generic stop words and named entities. The importance of stop words and named entities identification in TETSC is described earlier, and is applied in two stages. First, stop words and named entities that can occur in all text snippets are identified. For tweets these could for instance be locations and generic stop words such as 'the', 'a', and 'and'. Second, stop words and named entities are identified after identifying specific TEXT SNIPPET TYPES, which allows for the identification of stop words and named entities that are only important in, or occur in, specific TEXT SNIPPET TYPES.

Prior to being able to identify these type-specific stop words and named entities, however, the TEXT SNIPPET TYPES themselves need to be identified. In the case of tweets these types could be normal tweets, retweets or responses to tweets. Specific characteristics for these TEXT SNIPPET TYPES are to be identified, based on which rules for their identification can be created.

Once a specific TEXT SNIPPET TYPE is identified and is cleaned using both the generic and type-specific STOP WORD AND NAMED ENTITY LIST, an ENRICHMENT STRATEGY for this specific TEXT SNIPPET TYPE can be developed. This ENRICHMENT STRATEGY is related to the purpose of classification, and could for instance entail the original tweet a tweet is the response to.

When for every TEXT SNIPPET TYPE an ENRICHMENT STRATEGY is or is not developed, as not enriching a TEXT SNIPPET TYPE is also seen as an ENRICHMENT STRATEGY, the method continues to the model building activity. This entails cleaning the TEXT SNIPPET CORPUS using the previously defined STOP WORD AND NAMED ENTITY LIST, creating a CLASSIFIER based on this corpus of CLEAN TEXT SNIPPETS, creating TOPIC MODELS of the various EXTERNAL DATA SOURCES involved in the ENRICHMENT STRATEGIES, deploying the ENRICHMENT STRATEGIES through enriching TEXT SNIPPETS, and finally training another CLASSIFIER based on this newly created corpus of ENRICHED TEXT SNIPPETS.

Once the model building activity is performed everything is in place for classification, and the implementation of TETSC can be tested and evaluated.

3. IMPLEMENTATION: THE CASE OF FINANCIAL TRANSACTION DESCRIPTIONS

In this paper TETSC is evaluated through an implementation at a West-European company providing personal finance software. This company allows users to import their financial transactions, upon which the transactions are assigned a category using their name (max. 70 characters) and description (max. 140 characters). Based on the assignment of these categories users are then provided with an overview of their finances. Key in this process is that this categorization process is performed well. Therefore TETSC is implemented at this company.

3.1. Data Exploration

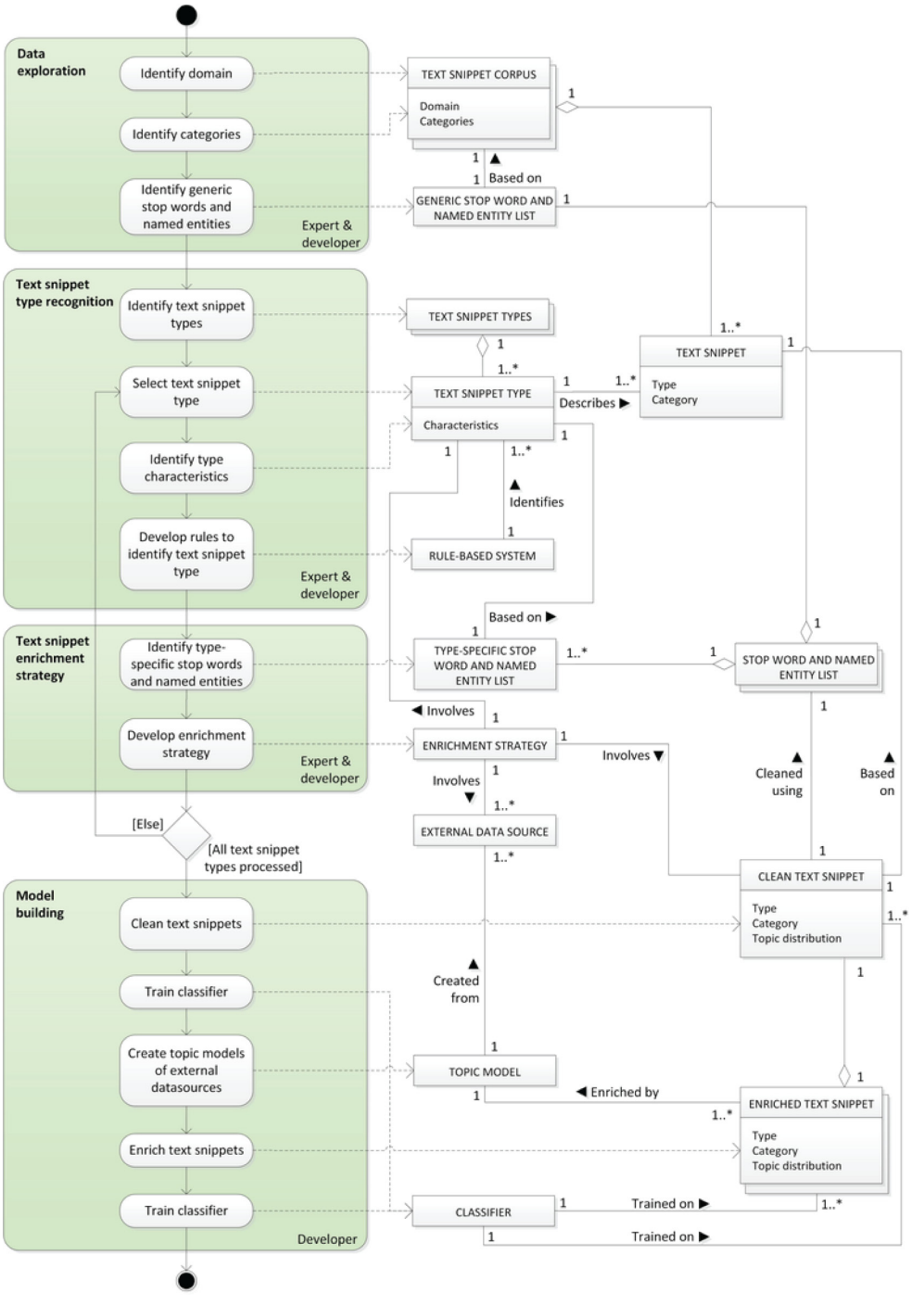
The domain is defined as 'Personal Finance', as both Business-to-Consumers (B2C) and Consumer-to-Consumer (C2C) transactions are processed. 76 distinct categories are identified. A plethora of generic named entities are identified, amongst which city names, dates, times, IBAN account numbers, and so forth. These named entities are identified through a handcrafted rule-based system, which proves successful because for all transactions the name field, and for B2C transactions also the description field, are generated through automated means, which causes common structures.

Preprocessing transactions using the GENERIC STOP WORD AND NAMED ENTITY LIST is an important step, even if only to improve similarity amongst (1) transactions imported from different banks, who are all free to present the same transactions in different ways, (2) transactions imported at different points in time, as for instance the enforced adoption of SEPA in Europe changes the way transactions are presented to consumers, and (3) transactions originating from different vendors, who to some extent are free to decide what is presented to consumers.

3.2. Text Snippet Type Recognition

Now generic stop words and named entities are identified for financial transactions, a look is taken at the different types of transactions. In this case they are identified as personal transactions, company transactions and undefined transactions. Hereby the type personal indicates C2C transactions, the type company indicates B2C transactions, and the type undefined indicates transactions where scriptural money is converted to a physical currency. Hereby the name

Figure 1. The topically-enriched text snippet classification (TETSC) method



'undefined' is chosen as when these transactions are imported it is unknown what the money is spent on exactly.

Next per type characteristics are identified, and rules are developed for their recognition. Hereby both negative and positive association are allowed. The reason for this is that it is hard to define personal transactions, other than on characteristics they usually do not contain, such as numeric sequences of more than 10 characters. Characteristics are easier to identify for company and undefined transactions, as these typically are more structured. When withdrawing money at a bank, for instance, a common structure is the word ATM, followed by a date, time and pass number.

3.3. Text Snippet Enrichment Strategy

Once a specific transaction type can be recognized, type-specific stop words and named entities are identified, which includes named-entities that are used by the consecutively developed enrichment strategies. For personal transactions the person's name is removed, as it might contain an existing word that influences enrichment while not being related to the purpose of the transaction at all. Note, however, that this person name is added again after enriching, as the person a transaction is made to can be quite identifying for a transaction. One might, for instance, play squash with the same person every week, making it more likely that a transaction to that person involves sports.

For company transactions the company name is identified based on heuristics. Firstly the transaction is inspected for the occurrence of Dutch variations on Ltd. or Inc. If these are not found, the name of a transaction is inspected. Through inspecting data from test accounts at different banks it becomes apparent that if the name of a transaction does not start with a previously identified named entity, it contains the company name. If the name does start with a named entity, however, the company name is most likely in the description field, and identification becomes harder. The description field is inspected for the occurrence of a Dutch city, one of the named entities identified at the generic stop words and named entities identification step. The reason for this is that large companies usually have multiple offices, whereby the city name is often added as an office identifier. If a city is not found either, the company name is determined to be the longest sequence of non-numerical characters in the description, containing at most 2 spaces.

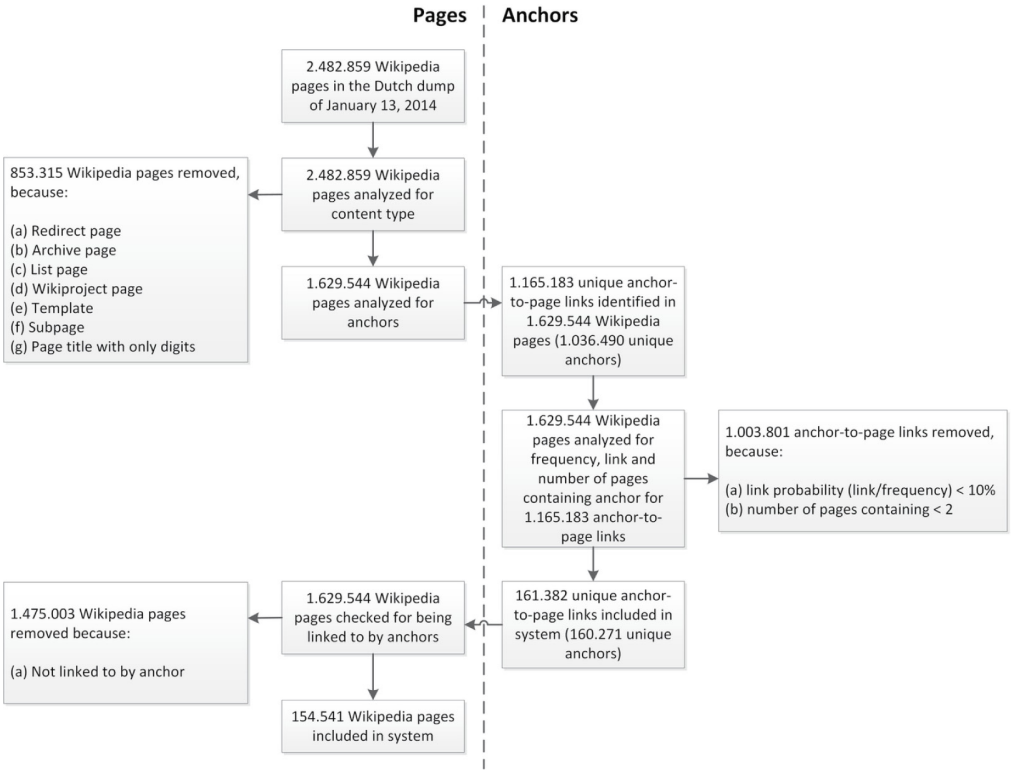
An enrichment strategy is then created for both personal and company transactions. No enrichment strategy is created for undefined transactions, as for these transactions the real category cannot be defined. The enrichment strategies are described per type in the following Subsections.

3.3.1. Personal Enrichment Strategy

For personal transactions the transactions are enriched using Wikipedia pages. The Dutch Wikipedia dump of 13 January 2014 is imported locally for this purpose using MediaWiki. Transactions are linked to Wikipedia pages using a replication of the Tagme method of Ferragina & Scaiella (2012). The results of the different steps are summarized and presented in Figure 2. Because of the unexpected low amount of remaining Wikipedia pages, the evaluation of anchors and pages by Tagme is taken into doubt and another implementation is created using the same method of linking transactions to Wikipedia pages, but without removing any anchors or pages. This method is indicated by 'no Tagme' in the results Section.

Of the Wikipedia pages of both methods a topic model is created using Gensim, a program that uses very little memory during model generation because the files the model is being created from are kept on disk instead of in-memory (Řehůřek & Sojka, 2010).

Figure 2. Summarized results of the implementation of Tagme on the Dutch Wikipedia



Every anchor name, the probability it links to a specific page, the probability it links to a page at all, the number of pages this anchor occurs on and the topic distribution of the full content of the Wikipedia page this anchor links to is then indexed in SOLR, a Lucene based search engine. Transactions are linked to anchors which in turn are linked to topic distributions of Wikipedia pages using queries containing transaction sequences of up to four words. Hereby sequences overlapping the name and description field are not allowed. See Figure 3 in Subsection 3.4 for an example.

3.3.2. Company Enrichment Strategy

Company transactions are enriched using the sector and trade of the company name identified earlier. Queries of the identified company names are sent to an online Dutch database containing all company names. This database contains all company names, as every Dutch company is enforced by law to register there. As the identified company name might contain information not included in this database, such as a (sometimes incomplete) city name, the query is reduced by one word if a company is not found.

Once a company is identified its unique identifier is recorded, and three other sources are queried for the sector and the trade of the company. First, a locally available database containing 2 million company names is queried. If the sector and trade are not found, two other websites specialized in the collection of company information are queried. Using this technique 82% of the identified companies (143,939 out of 176,229) are assigned a trade and sector. The reason

this percentage is not 100% is that the trades and sectors in these databases are mostly maintained manually, and therefore are not complete. Furthermore, some companies identified do not exist anymore and therefore might be skipped by such sites.

Linking identified company names to companies is done in an efficient way. If no corresponding company is found, this is recorded. If there is a corresponding company found, this too is recorded. Because of this the required amount of web requests decreases as the dataset grows.

3.4. Model Building

Based on a large dataset a training set of 10 million transactions is randomly generated and anonymized through removing transaction dates, amounts and account numbers. This training set is cleaned, of which the type division is shown in Table 1. Hereby removed indicates transactions removed from the training set because neither the name or description field contained any text. For company transactions approximately 500,000 probable company name Strings are identified, resulting in 176,229 identified companies as described in Subsection 3.3.2. The reason only 180 thousand companies are identified is because multiple probable companies can point toward the same identified company, as well as that some company names are maimed in transactions, and that some are simply identified wrongly.

The Maximum Entropy (MaxEnt) implementation of OpenNLP is used as a classifier, for it is often used in literature in combination with text snippets (*e.g.* Chen *et al.* 2011; Phan *et al.* 2008; Ritter et al. 2011). A classifier is trained on the training set before it is cleaned as a baseline and another after it is cleaned.

Two topic models are created of Wikipedia, as described in Subsection 3.3.1, and one topic model is created of the trade and sector of 143,939 companies. Because MaxEnt does not work with probabilities, whenever topic distributions are inferred, they are discretized using the method in Phan *et al.* (2008) and Nguyen *et al.* (2009) with probability intervals of 0.05 for the company topic model and 0.06 for the Wikipedia topic models.

As stated in Subsection 3.3.1 the topic distributions of Wikipedia pages are indexed in SOLR, and appended to transactions as indicated by Figure 3. When enriching first a query towards SOLR is generated, and of the returning results the discretized topic distributions are appended. At most one Wikipedia page is allowed as enrichment per four words in a transaction.

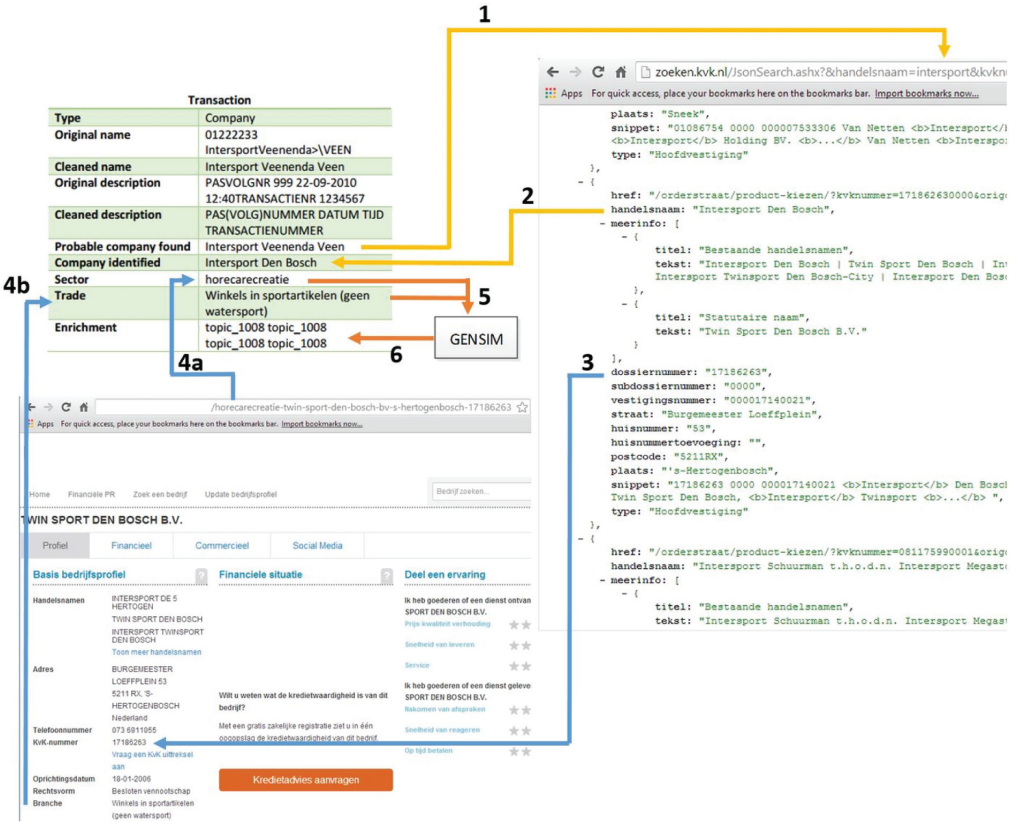
For company transactions the entire process described in Subsection 3.3.2 is performed. Hereby company identification and topic inference for new transactions is performed real-time. See Figure 4 for a possible scenario. First the company name is identified, and queried at the Dutch database containing all company names (1). A company is identified (2), but no entry in the local database exists for this company. Therefore an external site is queried for this company (3), and the trade and sector are identified (4a + 4b). This trade and sector are then sent to Gensim (5), and the inferred topic distribution is returned (6). Hereby company transaction topic numbers are increased by 1,000 to prevent interference with topics generated for personal transactions.

Table 1. Transaction type distribution for the training set

Transaction Type				
Personal	**Company**	**Undefined**	**Total**	**Removed**
972,536	8,196,554	776,279	9,945,369	54,631

Figure 3. Personal transaction enrichment visualized. The transaction is fictitious, based on one of the researchers.

Figure 4. Company transaction enrichment visualized. The transaction is fictitious, based on one of the researchers.



Multiple classifiers are trained on the transactions with varying combinations of the enrichment methods, as is described in the results Section. The resulting classification architecture of the implementation of TETSC can be found in Figure 5.

4. RESULTS: TOWARDS ERROR AND DIMENSIONALITY REDUCTIONS

In order to test the implementation of TETSC a test set of 5,313 transactions is created and manually verified. The type distribution of this test set is shown in Table 2. Of this test set variations corresponding to the created training sets are created. Tests are performed on a laptop, with hard- and software specifications as described in Table 3.

The results of the initial tests are shown in Table 4. Additionally in this Table is the 'Sanitized' set, for which the preprocessing technique based purely on stop word removal in place at the personal finance company prior to implementing TETSC is used. Shown here is that the initial classification performance without enrichment is already quite high, and that using the different enrichment methods seemingly little classification performance increase is achieved when looking purely at the $F_{0,25}$-score.

Figure 5. Classification architecture after implementing TETSC

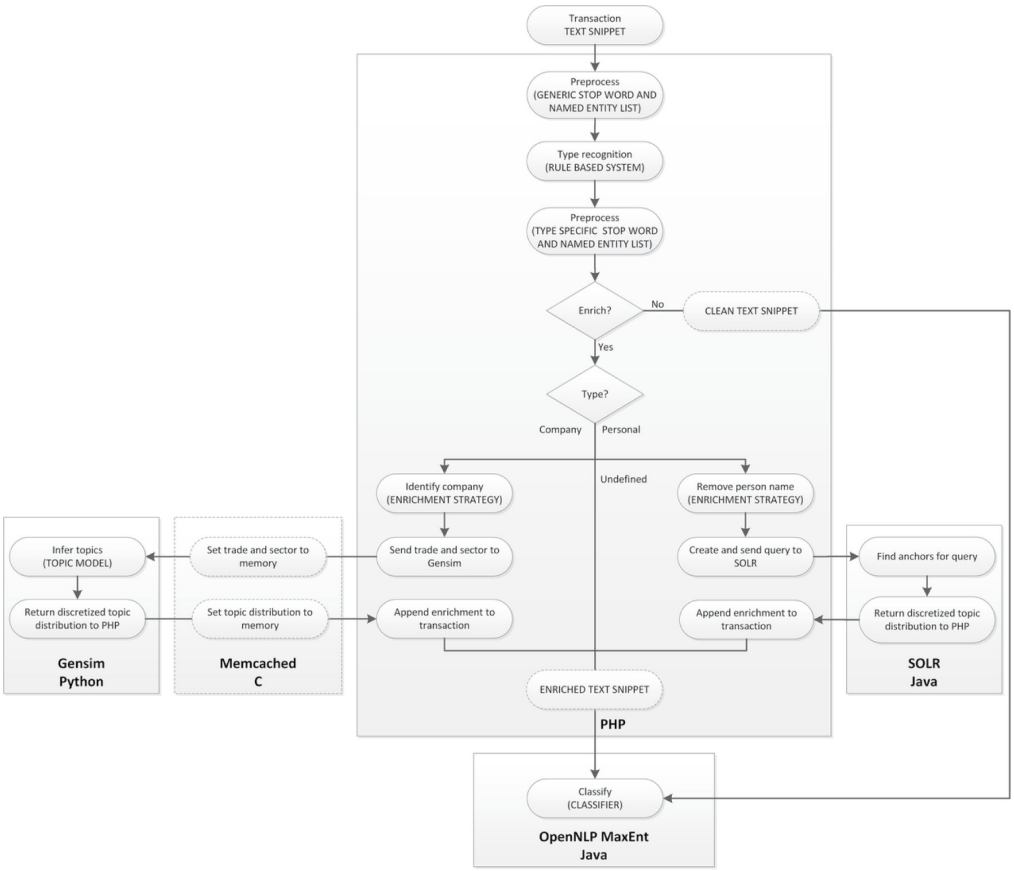


Table 2. Transaction type distribution for the test set

Transaction Type			
Personal	**Company**	**Undefined**	**Total**
415	4,457	441	5,313

Table 3. Hard and software specifications of the system where performance is tested on

Specification	Value
Model	Dell Latitude E6530
Processor	Intel i7-3740QM (2,7/3,7 GHz)
RAM	16GB DDR3 (1600 MHz)
Storage	ADATA SSD SX900 512GB-DL2
PHP	64-bit, version 5.4.12
MySQL	Version 5.6.12
Java	64-bit, version 1.7.0 u45
Python	WinPython 64-bit, version 2.7.6.3
SOLR	Version 4.7.0
Memcached (server)	Version 1.4.5_4_gaa7839e
OpenNLP	Version 1.5.3
Gensim	Version 0.9.1

Interesting to see is that the 'sanitized' set performs really well for personal transactions, but worse for the other types transactions. Because less than 10% of the transactions are of the type personal, the overall performance increase of the sanitized type is marginal.

Furthermore, interesting to see is that employing only the enrichment strategy for personal transactions performs better than employing both the personal and company enrichment strategies together. This, while both strategies individually increase performance when compared to not employing any enrichment strategy at all.

The time required by and the dimensionality of the different training sets is shown in Table 5. Interesting to see here is that the 'clean' set implemented from TETSC performs a lot better in dimensionality reduction than the 'sanitized' set. Furthermore, whereas the 'clean' set requires more processing time, thereby impairing efficiency, this can be accounted to the recognition of city names in transactions.

Based on the initial results additional tests are run, with varying parameters for the topic models and the classifiers. A best-of-breed model is created, of which the performance is shown in Table 6. This shows that using the best combination of sets an $F_{0.25}$-score of 88.71% is achieved, which is an error reduction of 21.26% from using no preprocessing method or enrichment strategy at all.

The best-of-breed model requires the usage of different preprocessing methods for transaction types in the generation of classification models for specific transaction types. These are shown in Table 7. It might seem odd that transactions are preprocessed in different ways for classification model generation. Since these models do not have to be generated real-time, however, this incurs no efficiency decrease.

Table 4. $F_{0.25}$-score for the different training sets. Topic models are created using k = 150, α = 0.3 and η = 0.01. Topic discretization intervals for company transactions are 0.05, and 0.06 for personal transactions. MaxEnt models are generated using 250 iterations.

Training Set	F0,25-Score			
	Total	Personal	Company	Undefined
Dirty	85.66%	60.48%	87.55%	**90.25%**
Sanitized	85.77%	**79.47%**	86.02%	89.12%
Clean	86.41%	62.17%	88.36%	89.57%
Clean (w/o city recognition)	86.41%	62.17%	88.36%	89.57%
Clean + Topics personal (Tagme)	86.35%	61.20%	88.38%	89.57%
Clean + Topics personal (no Tagme)	**86.75%**	62.89%	**88.67%**	89.80%
Clean + Topics companies	86.50%	61.20%	88.56%	89.57%
Clean + Topics companies, personal (Tagme)	86.54%	61.45%	88.58%	89.57%
Clean + Topics companies, personal (no Tagme)	86.66%	62.41%	88.60%	89.80%

Table 5. Time requirements of the different steps of TETSC for the training sets, and the dimensionality of their data

Training Set	Clean Time (in Seconds)	Clean Time (Transactions per Second)	Classification Time (in Seconds)	Number of Terms
Dirty	1	5,313	<1	8,513,393
Sanitized	12	433	<1	1,871,512
Clean	38	140	<1	824,617
Clean (w/o city recognition)	13	409	<1	824,617
Clean + topics personal (Tagme)	38,5	138	<1	824,767
Clean + topics personal (no Tagme)	39	136	<1	824,767
Clean + topics companies	40	133	<1	823,503
Clean + topics companies, personal (Tagme)	40	133	<1	825,220
Clean + topics companies, personal (no Tagme)	43	124	<1	825,220

*Table 6. $F_{0.25}$-score and error reduction of the best-of-breed model created for classification using TETSC. * Error reduction from using no preprocessing or enrichment method at all.*

	Personal	Company	Undefined	Total
$F_{0.25}$-score	87.23%	88.67%	90.48%	88.71%
Error reduction*	67.68%	9.01%	2.33%	21.26%

Table 7. The required preprocessing and enrichment methods for specific transaction types (columns), as is required to create the classification models that achieve the best classification performance in classifying specific transaction types (rows). Topic models are created using α = 1 / k and η = 0.01. The topic discretization interval for company transactions are 0.05, and 0.06 for personal transactions. MaxEnt models are generated using 500, 250 and 50 iterations for the models respectively from top to bottom.

	Preprocessing Method		
Transaction Type	**Personal**	**Company**	**Undefined**
Personal	Sanitized + topics personal (no Tagme, k = 50)	Clean	Clean
Company	Clean + topics personal (no Tagme, k = 150)	Clean	Clean
Undefined	Dirty	Dirty	Dirty

5. CONCLUSION

This paper presents TETSC, the Topically-Enriched Text Snippet Classification method. Through implementation TETSC shows flexible in allowing the identification of multiple text snippet types and enrichment strategies for said types. Whereas it is shown that classification performance for financial transactions without applying any preprocessing or enrichment is already high, it is also shown that the error can reduced by an additional 21% through stop word removal, NER and topical enrichment, while simultaneously reducing dimensionality by 90%.

Furthermore, this paper shows a practical implementation of the topic probability distribution discretization method of Phan *et al.* (2008) and Nguyen *et al.* (2009), the partial implementation of the Tagme method of Ferragina & Scaiella (2012) and the meta-modeling technique of Weerd & Brinkkemper (2008). Whereas application of the TETSC method does not permanently solve the problem of correctly classifying domain-specific text snippet to predefined categories, it certainly contributes towards this solution by both applying existing knowledge contained in the scientific knowledge base, as well as adding new.

The main limitation of this paper is generalizability. TETSC is validated through only one implementation. While good results are achieved, the question remains if such improvements are reproducible for text snippets in other domains.

Table 8. Glossary

Term	Definition
ATM	Automated Teller Machine
B2C	Business-to-Consumer
C2C	Consumer-to-Consumer
MaxEnt	Maximum Entropy
NER	Named-Entity Recognition
PDD	Process-Deliverable Diagram
TETSC	Topically-Enriched Text Snippet Classification method

REFERENCES

Blei, D. M., & McAuliffe, J. D. (2007). In J. C. Plaat, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *Supervised topic models* (Vol. 20, pp. 121–128). Advances in Neural Information Processing Systems Vancouver, British Columbia, Canada: Curran Associates, Inc.

Borthwick, A. (1999). *A Maximum Entropy Approach to Named Entity Recognition* (Doctoral dissertation). New York University:NY.

Carpineto, C., & Romano, G. (2012). A Survey of Automatic Query Expansion in Information Retrieval. *ACM Computing Surveys*, *44*(1), 1–50. doi:10.1145/2071389.2071390

Chen, M., Xiaoming, J., & Shen, D. (2011). Short Text Classification Improved by Learning Multi-Granularity Topics. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Barcelona, Catalonia, Spain, 1776-1781.

Domingos, P., & Pazzani, M. (1997). On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. *Machine Learning*, *29*(2-3), 103–130. doi:10.1023/A:1007413511361

Dragut, E., Fang, F., Sistla, P., Yu, C., & Meng, W. (2009). Stop word and related problems in web interface integration. *Proceedings of the VLDB Endowment*, *2*(1), 349–360. doi:10.14778/1687627.1687667

Ferragina, P., & Scaiella, U. (2012). Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Software*, *29*(1), 70–75. doi:10.1109/MS.2011.122

Gabrilovich, E., & Markovitch, S. (2005). Feature Generation for Text Categorization Using World Knowledge. *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, UK, 1048-1053.

Gabrilovich, E., & Markovitch, S. (2006). Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge Feature Generation with Wikipedia. *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, Massachusetts, USA, 1301-1306.

Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, *28*(9), 921–932. doi:10.1145/4284.4286

Hu, X., Sun, N., Zhang, C., & Chua, T.-S. (2009). Exploiting internal and external semantics for the clustering of short texts using world knowledge. *Proceeding of the 18th ACM conference on Information and knowledge management*, Hong Kong, China, 919-929. doi:10.1145/1645953.1646071

Nguyen, C., Phan, X., Horiguchi, S., Nguyen, T.-T., & Ha, Q.-T. (2009). Web Search Clustering and Labeling with Hidden Topics. [TALIP]. *ACM Transactions on Asian Language Information Processing*, *8*(3), 12–52. doi:10.1145/1568292.1568295

Otten, S., & Spruit, M. (2011). *Linguistic engineering and its applicability to business intelligence: towards an integrated framework*. International Conference on Knowledge Discovery and Information Retrieval (pp. 460–464). Paris, France: SciTePress.

Pachidi, S., Spruit, M., & van der Weerd, I. (2014). Understanding Users' Behavior with Software Operation Data Mining. *Computers in Human Behavior*, *30*, 583–594. doi:10.1016/j.chb.2013.07.049

Phan, X.-H., Nguyen, L.-M., & Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *Proceeding of the 17th international conference on World Wide* Web, Beijing, China, 91-100). New York, New York, USA: ACM Press. doi:10.1145/1367497.1367510

Ramage, D., Dumais, S., & Liebling, D. (2010). Characterizing Microblogs with Topic Models. *Proceedings of the 4th International Conference on Weblogs and Social Media*, Washington DC, DC, USA.

Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the 2010 Language Resources and Evaluation Workshop on New Challenges for NLP Frameworks*, Valletta, Malta, 45-50.

Ritter, A., Clark, S., & Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK, 1524-1534.

Sahami, M., & Heilman, T. D. (2006). A web-based kernel function for measuring the similarity of short text snippets. *Proceedings of the 15th international conference on World Wide Web*, Edinburgh, Scotland, UK, 377-386. doi:10.1145/1135777.1135834

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. [CSUR]. *ACM Computing Surveys*, *34*(1), 1–47. doi:10.1145/505282.505283

Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J., & Yang, Q. (2006). Query enrichment for web-query classification. *ACM Transactions on Information Systems*, *24*(3), 320–352. doi:10.1145/1165774.1165776

Simske, S. (2013). *Meta-Algorithmics: Patterns for Robust, Low-Cost, High-Quality Systems*. Oxford, UK: Wiley. doi:10.1002/9781118626719

Spruit, M., Vroon, R., & Batenburg, R. (2014). Towards healthcare business intelligence in long-term care: An explorative case study in the Netherlands. *Computers in Human Behavior*, *30*, 698–707. doi:10.1016/j.chb.2013.07.038

Tsukayama, H. (2013). Twitter turns 7: Users send over 400 million tweets per day. *Washington Post*. Retrieved November 19, 2013, from: http://articles.washingtonpost.com/2013-03-21/business/37889387_1_tweets-jack-dorsey-twitter

Twitter Engineering. (2011). 200 million Tweets per day. *Twitter*. Retrieved November 19, 2013, from: https://blog.twitter.com/2011/200-million-tweets-day

van de Weerd, I., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. In M. R. Syed & S. N. Syed (Eds.), *Handbook of research on modern systems analysis and design technologies and applications* (pp. 38–58). United Kingdom: Information Science Reference. doi:10.4018/978-1-59904-887-1.ch003

Vleugel, A., Spruit, M., & van Daal, A. (2010). Historical data analysis through data mining from an outsourcing perspective: The three-phases method. *International Journal of Business Intelligence Research*, *1*(3), 42–65. doi:10.4018/jbir.2010070104

Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using Wikipedia. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Las Vegas, Nevada, USA, 713-721. doi:10.1145/1401890.1401976

Marco Spruit is a researcher at the Information and Computing Sciences Department of Utrecht University in the Netherlands. Marco's research interest is Decision Analytics for Transparancy (DAT) with special attention to healthcare applications. Marco serves on the editorial board for the international journals on Business Intelligence Research (IJBIR) and Decision Analytics, and Computer Information Systems (JCIS). Before 2007 Marco worked in industry as a software developer for fourteen years in the fields of Business Intelligence and Text Analytics.

Bas Vlug works in industry as a software architect in the field of Enterprise Resource Planning. Prior to that Bas worked as a developer at a personal finance company, after completing his Master's degree in Business Informatics at Utrecht University in the Netherlands.

APPENDIX A

Table 9. Concepts of TETSC

Concept	Description
text snippet corpus	A corpus is a collection of documents (Sebastiani, 2002). A TEXT SNIPPET CORPUS is a corpus of TEXT SNIPPETS.
text snippet	A small section of text that typically does not contain more than two or three sentences.
text snippet types	The collection of all types of TEXT SNIPPETS for this TEXT SNIPPET CORPUS.
text snippet type	A singular type of TEXT SNIPPET. A TEXT SNIPPET TYPE has characteristics that allow the type to be recognized.
rule-based system	A modularized know-how system, where know-how is practical problem-solving knowledge (Hayes-Roth, 1985). In TETSC knowledge of TEXT SNIPPETS is used to recognize TEXT SNIPPET TYPES.
stop word and named entity list	Stop words are words that do not convey any significant semantics to the texts or phrases they appear in (Dragut *et al.*, 2009). A named entity is a form of information extraction in which we seek to classify every word in a document as being a person-name, organization, location, date, time, monetary value, percentage, or "none of the above" (Borthwick, 1999). A STOP WORD AND NAMED ENTITY LIST is a list of stop words to be removed and named entities to be recognized and consists of a GENERIC STOP WORD AND NAMED ENTITY LIST and multiple TYPE SPECIFIC STOP WORD AND NAMED ENTITY LIST.
generic stop word and named entity list	Generic stop words and named entities are words and named entities that commonly occur amongst all TEXT SNIPPETS, and a GENERIC STOP WORD AND NAMED ENTITY LIST is a list thereof.
type-specific stop word and named entity list	Type-specific stop words and named entities are words and named entities that commonly occur only amongst one TEXT SNIPPET TYPE. A TYPE SPECIFIC STOP WORD AND NAMED ENTITY LIST is a list thereof.
enrichment strategy	Enrichment is defined as "improving or enhancing the quality or value of" (The Oxford Dictionary). ENRICHMENT STRATEGY is the strategy of how this improvement or enhancement of the quality or value of a specific TEXT SNIPPET TYPE is achieved. An ENRICHMENT STRATEGY is TEXT SNIPPET TYPE-specific and involves EXTERNAL DATA SOURCE(S) and CLEAN TEXT SNIPPETS.
external data source	A data source other than the TEXT SNIPPET CORPUS. An EXTERNAL DATA SOURCE can be either on- or offline available.
topic model	A TOPIC MODEL is a model of topics, where a topic is a probability distribution over terms in a vocabulary (Blei & McAuliffe, 2007). TOPIC MODELS in TETSC are created from EXTERNAL DATA SOURCE(S) and are used to enrich CLEAN TEXT SNIPPETS based on an ENRICHMENT STRATEGY.
clean text snippet	A TEXT SNIPPET with stop words removed and named entities recognized using the STOP WORD AND NAMED ENTITY LIST.
enriched text snippet	A CLEAN TEXT SNIPPET enriched by topics from a TOPIC MODEL, as described by the ENRICHMENT STRATEGY.
classifier	A CLASSIFIER takes an unlabeled example and assigns it to a class (Domingos & Pazzani, 1997). In TETSC two CLASSIFIERS are created, one for classifying CLEAN TEXT SNIPPETS and one for classifying ENRICHED TEXT SNIPPETS.

APPENDIX B

Table 10. Activities of TETSC

Activity	Sub-Activity	Description
Data exploration	Identify domain	The TEXT SNIPPET CORPUS used in TETSC has a domain which is to be identified.
	Identify categories	TEXT SNIPPETS in the TEXT SNIPPET CORPUS are to be assigned to categories. These categories need to be identified.
	Identify generic stop words and named entities	TEXT SNIPPETS contain generic stop words that provide no added value and can be filtered out of or replaced in every TEXT SNIPPET. Aside from this TEXT SNIPPETS contain named entities which can be recognized. These potential generic stop words and named entities are to be identified and added to the GENERIC STOP WORD AND NAMED ENTITY LIST.
Text snippet type recognition	Identify text snippet types	TEXT SNIPPETS in the TEXT SNIPPET CORPUS may be of different types requiring different preprocessing and enrichment techniques. These types are to be identified.
	Select text snippet type	One single TEXT SNIPPET TYPE is to be selected, for which the following activities are to be performed.
	Identify type characteristics	A TEXT SNIPPET TYPE has characteristics which allow for the recognition of the type in a snippet. These characteristics are to be identified.
	Develop rules to identify text snippet type	The characteristics of the TEXT SNIPPET TYPE can be automatically recognized using rules. These rules are to be developed and added to the RULE-BASED SYSTEM.
Text snippet enrichment strategy	Identify type-specific stop words and named entities	Aside from generic stop words and named entities, a TEXT SNIPPET TYPE contains type-specific stop words that should be removed or replaced and named entities that should be recognized in the case of a specific TEXT SNIPPET TYPE. These potential stop words and named entities are added to a TYPE-SPECIFIC STOP WORD AND NAMED ENTITY LIST.
	Develop enrichment strategy	Identified characteristics and named entities of the TEXT SNIPPET TYPE reveal information that can be used to devise a type-specific ENRICHMENT STRATEGY using EXTERNAL DATA SOURCE(S). This enrichment strategy is to be developed.
Model building	Clean text snippets	Having identified stop words and named-entities for all TEXT SNIPPET TYPES, the next step is to clean the TEXT SNIPPET CORPUS using the STOP WORD AND NAMED ENTITY LIST.
	Train classifier	A CLASSIFIER is to be trained on (a training set of) the TEXT SNIPPET CORPUS containing CLEAN TEXT SNIPPETS or ENRICHED TEXT SNIPPETS.
	Create topic models of external data sources	TOPIC MODELS are created of EXTERNAL DATA SOURCES used in ENRICHMENT STRATEGIES.
	Enrich text snippets	CLEAN TEXT SNIPPETS are enriched using topics of the TOPIC MODELS. These topics are inferred from enrichments gathered from EXTERNAL DATA SOURCES as described in the ENRICHMENT STRATEGY.