How to Distinguish a Kidney Theft from a Death Car? Experiments in Clustering Urban-Legend Texts

Roman Grundkiewicz, Filip Graliński
Adam Mickiewicz University
Faculty of Mathematics and Computer Science
Poznań, Poland
rgrundkiewicz@amu.edu.pl, filipg@amu.edu.pl

Abstract

This paper discusses a system for automatic clustering of urban-legend texts. Urban legend (UL) is a short story set in the present day, believed by its tellers to be true and spreading spontaneously from person to person. A corpus of Polish UL texts was collected from message boards and blogs. Each text was manually assigned to one story type. The aim of the presented system is to reconstruct the manual grouping of texts. It turned out that automatic clustering of UL texts is feasible but it requires techniques different from the ones used for clustering e.g. news articles.

1 Introduction

Urban legend is a short story set in the present day, believed by its tellers to be true and spreading spontaneously from person to person, often including elements of humour, moralizing or horror. Urban legends are a form of modern folklore, just as traditional folk tales were a form of traditional folklore, no wonder they are of great interest to folklorists and other social scientists (Brunvand, 1981). As urban legends (and related rumours) often convey misinformation with regard to controversial issues, they can draw the attention of general public as well[1].

The traditional way of collecting urban legends was to interview informants, to tape-record their narrations and to transcribe the recordings (Brunvand, 1981). With the exponential growth of the Internet, more and more urban-legend texts turn up on message boards or blogs and in social media in general. As the web circulation of legends is much easier to tap than the oral one, it becomes feasible to envisage a system for the machine identification and collection of urban-legend texts. In this paper, we discuss the first steps into the creation of such a system. We concentrate on the task of clustering of urban legends, trying to reproduce automatically the results of manual categorisation of urban-legend texts done by folklorists.

In Sec. 2 a corpus of 697 Polish urban-legend texts is presented. The techniques used in preprocessing the corpus texts are discussed in Sec. 3, whereas the clustering process – in Sec. 4. We present the clustering experiment in Sec. 5 and the results – in Sec. 6.

2 Corpus

The corpus of $N = 697$ Polish urban-legend texts was manually collected from the Web, mainly from message boards and blogs[2]. The corpus was not gathered with the experiments of this study in mind, but rather for the purposes of a web-site dedicated to the collection and documentation of the Polish web folklore[3]. The following techniques were used for the extraction of web pages with urban legends:

1. querying Google search engine with formulaic expressions typical of the genre of urban legends, like *znajomy znajomego* (= *friend of a friend*), *słyszałem taką opowieść* (= *I heard this story*) (Graliński, 2009),

2. given a particular story type and its examples, querying the search engine with various combinations of the keywords specific for the story type, their synonyms and paraphrases,

3. collecting texts and links submitted by readers of the web-site mentioned above,

[1] Snopes.com, an urban legends reference web-site, is ranked #2,650 worldwide by Alexa.com (as of May 3, 2011), see http://www.alexa.com/siteinfo/snopes.com

[2] The corpus is available at http://amu.edu.pl/~filipg/uls.tar.gz

[3] See http://atrapa.net

Proceedings of the Workshop on Information Extraction and Knowledge Acquisition, pages 29–36,
Hissar, Bulgaria, 16 September 2011.

4. analysing backlinks to the web-site mentioned above (a message board post containing an urban legend post is sometimes accompanied by a reply "it was debunked here: [link]" or similar),

5. collecting urban-legend texts occurring on the same web page as a text found using the methods (1)-(4) – e.g. a substantial number of threads like "tell an interesting story", "which urban legends do you know?" or similar were found on various message boards.

The web pages containing urban legends were saved and organised with Firefox Scrapbook add-on[4].

Admittedly, method (2) may be favourable to clustering algorithms. This method, however, accounted for about 30% of texts [5] and, what's more, the synonyms and paraphrases were prepared manually (without using any lexicons), some of them being probably difficult to track by clustering algorithms.

Urban-legend texts were manually categorised into 62 story types, mostly according to (Brunvand, 2002) with the exception of a few Polish legends unknown in the United States. The number of texts in each group varied from 1 to 37.

For the purposes of the experiments described in this paper, each urban legend text was manually delimited and marked. Usually a whole message board or blog post was marked, but sometimes (e.g. when an urban legend was just quoted in a longer post) the selection had to be narrowed down to one or two paragraphs. The problems of automatic text delimitation are disregarded in this paper.

Two sample urban-legend texts (both classified as the *kidney theft* story type) translated into English are provided below. The typographical and spelling errors of the original Polish texts are preserved in translation.

> Well yeah... the story maybe not too real, but that kids' organs are being stolen it's actually true!! More and more crimes of this kind have been reported, for example in the Lodz IKEA there was such a crime, to be more precse a girl was kidnapped from this place where parents leave their kids and go shopping (a mini kids play paradise).Yes the gril was brought back home but without a kidney... She is about 5 years old. And I know that becase this girl is my mom's colleague family... We cannot panic and hide our kids in the corners, or pass on such info via GG [Polish instant messenger] cause no this one's going believe, but we shold talk about such stuf, just as a warning...
>
> I don't know if you heard about it or not, but I will tell you one story which happened recently in Koszalin. In this city a very large shopping centre- Forum was opened some time ago. And as you know there are lots of people, commotion in such places. And it so happened that a couple with a kid (a girl, I guess she was 5,6 years old I don't know exactly) went missing. And you know they searched the shops etc themselves until in the end they called the police. And they thought was kidnapping, they say they waited for an information on a ransom when the girl was found barely alive without a kidney near Forum. Horrible...; It was a shock to me especially cause I live near Koszalin. I'm 15 myself and I have a little sister of a similar age and I don't know what I'd do if such a thing happened to her... Horrible...[6]

The two texts represents basically the same story of a kidney theft, but they differ considerably in detail and wordings.

A smaller subcorpus of 11 story types and 83 legend texts were used during the development.

Note that the corpus of urban-legend texts can be used for other purposes, e.g. as a story-level paraphrase corpus, as each time a given story is re-told by various people in their own words.

3 Document Representation

For our experiments, we used the standard Vector Space Model (VSM), in which each document

[4] http://amb.vis.ne.jp/mozilla/scrapbook/

[5] The exact percentage is difficult to establish, as information on which particular method led to which text was not-saved.

[6] The original Polish text: http://www.samosia.pl/pokaz/341160/jestem_przerazona_jak_mozna_komus_podwedzic_dziecko_i_wyciac_mu_nerke_w_ch

is represented as a vector in a multidimensional space. The selection of terms, each corresponding to a dimension, often depends on a distinctive nature of texts. In this section, we describe some text processing methods, the aim of which is to increase similarity of documents of related topics (i.e. urban legends of the same story type) and decrease similarity of documents of different topics.

3.1 Stop Words

A list of standard Polish stop words (function words, most frequent words etc.) was used during the normalization. The stop list was obtained by combining various resources available on the Internet[7]. The final stop list contained 642 words.

We decided to expand the stop list with some domain-specific and non-standard types of words.

3.1.1 Internet Slang Words

Most of the urban-legend texts were taken from message boards, no wonder Internet slang was used in many of them. Therefore the most popular slang abbreviations, emoticons and onomatopoeic expressions (e.g. *lol, rotfl, btw, xD, hahaha*) were added to the stoplist.

3.1.2 Abstract Verbs

Abstract verbs (i.e. verbs referring to abstract activities, states and concepts rather than to the manipulation of physical objects) seem to be irrelevant for the recognition of the story type of a given legend text. A list of 379 abstract verbs was created automatically using the lexicon of the Polish-English rule-based machine translation system Translatica[8] and taking the verbs with subordinate clauses specified in their valency frames. This way, verbs such as *mówić* (= *say*), *opowiadać* (= *tell*), *pytać* (= *to ask*), *decydować* (= *decide*) could be added to the stop list.

3.1.3 Unwanted Adverbs

A list of Polish intensifiers, quantifiers and sentence-level adverbs was taken from the same lexicon as for the abstract verbs. 536 adverbs were added to the stop list in this manner.

3.1.4 Genre-specific Words

Some words are likely to occur in any text of the given domain, regardless of the specific topic. For example in a mathematical text one can expect words like *function, theorem, equation,* etc. to occur, no matter which topic, i.e. branch of mathematics (algebra, geometry or mathematical analysis), is involved.

Construction of the domain keywords list based on words frequencies in the collection of documents may be insufficient. An external, human knowledge might be used for specifying such words. We decided to add the words specific to the genre of urban legends, such as:

- words expressing family and interpersonal relations, such as *znajomy* (= *friend*), *kolega* (= *colleague*), *kuzyn* (= *cousin*) (urban legends are usually claimed to happen to *a friend of a friend*, *a cousin of a colleague* etc.),
- words naming the genre of urban legends and similar genres, e.g. *legenda* (= *legend*), *anegdota* (= *anecdote*), *historia* (= *story*),
- words expressing the notions of authenticity or inauthenticity, e.g. *fakt* (= *fact*), *autentyczny* (= *authentic*), *prawdziwy* (= *real*), as they are crucial for the definition of the genre.

3.2 Spell Checking

As very informal style of communication is common on message boards and even blogs, a large number of typographical and spelling errors were found in the collected urban-legend texts. The Hunspell spell checker was used to find misspelled words and generate lists of correction suggestions. Unfortunately, the order in which Hunspell presents its suggestions is of no significance, and consequently it is not trivial to choose the right correction. We used the observation that it is quite likely for the right correction to occur in the corpus and we simply selected the Hunspell suggestion that is the most frequent in the whole corpus. This simple method turned out to be fast and good enough for our application.

3.3 Lemmatisation and stemming

For the lemmatisation and stemming *morfologik-stemming* package[9] was used. This tool is based on an extensive lexicon of Polish inflected forms (as Polish is a language of rather complex inflection rules, there is no simple stemming algorithm as effective as Porter's algorithm for English (Porter, 1980).)

[7] http://www.ranks.nl/stopwords/polish.html, http://pl.wikipedia.org/wiki/Wikipedia:Stopwords

[8] http://poleng.pl/poleng/en/node/597

[9] http://morfologik.blogspot.com/

3.4 Use of Thesaurus

A thesaurus of synonyms and near-synonyms might be used in order to increase the quality of the distance measure between documents. However, in case of polysemous words word-sense disambiguation would be required. As no WSD system for Polish was available we decided to adopt a naive approach of constructing a smaller thesaurus containing only unambiguous words.

As the conversion of diminutives and augmentatives to forms from which they were derived can be regarded as a rather safe normalisation, i.e. there are not many problematic diminutives or augmentatives, such derivations were taken into account during the normalisation. Note that diminutive forms can be created for many Polish words (especially for nouns) and are very common in the colloquial language.

A list of Polish diminutives and augmentatives has been created from a dump of Wiktionary[10] pages. The whole list included above 5.5 thousand sets of words along with their diminutives and augmentatives.

4 Document Clustering

The task of document clustering consists in recognising topics in a document collection and dividing documents according to some similarity measure into K clusters. Representing documents in a multi-dimensional space makes it possible to use well-known general-purpose clustering algorithms.

4.1 Clustering Algorithms

K-Means (KM) (Jain et al., 1999; Berkhin, 2002; Manning et al., 2009) is the most widely used flat partitioning clustering algorithm. It seeks to minimise the average squared distances between objects in the same cluster:

$$RSS(K) = \sum_{k=1}^{K} \sum_{\vec{x}_i \in C_k} \|\vec{x}_i - \vec{c}_k\|^2 \qquad (1)$$

in subsequent iterations until a convergence criterion is met. The $\vec{x}_i$ value means the vector representing the ith document from collection, and $\vec{c}_k$ means the centroid of the kth cluster. There is, however, no guarantee that the global optimum is reached – the result depends on the selection of initial cluster centres (this issue is discussed in Sec. 4.2).

In all of our tests, K-Means turned out to be less efficient than the algorithm known as **K-Medoids** (KMd). K-Medoids uses medoids (the most centrally located objects of clusters) instead of centroids. This method is more robust to noise and outliers than K-Means. The simplest implementation involves the selection of a medoid as the document closest to the centroid of a given cluster.

We examined also popular agglomerative hierarchical clustering algorithms: **Complete Linkage** (CmpL), **Average Linkage** (AvL), known as UPGMA, and **Weighted Average Linkage** (Jain et al., 1999; Berkhin, 2002; Manning et al., 2009). These algorithms differ in how the distance between clusters is determined: in Complete Linkage it is the maximum distance between two documents included in the two groups being compared, whereas in Average Linkage – the average distance, whereas in the last one, distances are weighted based on the number of documents in each of them. It is often claimed that hierarchical methods produce better partitioning than flat methods (Manning et al., 2009). Other agglomerative hierarchical algorithms with various linkage criteria that we tested (i.e. Single Linkage, Centroid Linkage, Median Linkage and Ward Linkage), were outperformed by the ones described above.

We tested also other types of known clustering algorithms. Divisive hierarchical algorithm Bisecting K-Means and fuzzy algorithms as Fuzzy K-means, Fuzzy K-medoids and K-Harmonic Means were far less satisfactory. Moreover, in the case of fuzzy methods it is difficult to determine the fuzziness coefficient.

4.2 Finding the Optimal Seeds

One of the disadvantages of K-means algorithm is that it heavily depends on the selection of initial centroids. Furthermore, the random selection makes algorithm non-deterministic, which is not always desired. Many methods have been proposed for optimal seeds selection (Peterson et al., 2010).

One of the methods which can be used in order to select good seeds and improve flat clustering algorithms is **K-Means++** (KMpp) (Arthur and Vassilvitskii, 2007). Only the first cluster centre is selected uniformly at random in this method,

[10] Polish version of Wiktionary: `http://pl.wiktionary.org/wiki/`

each subsequent seed is chosen from among the remaining objects with probability proportional to the second power of its distance to its closest cluster centre. K-Means++ simply extends the standard K-Means algorithm with a more careful seeding schema, hence an analogous K-Medoids++ (KMdpp) algorithm can be easily created. Note that K-Means++ and K-Medoids++ are still non-deterministic.

We propose yet another approach to solving seeding problem, namely **centres selection by reduction of similarities (RS)**. The goal of this technique is to select the K-element subset with the highest overall dissimilarity. The reduction of similarities consists in the following steps:

1. Specify the number of initial cluster centres (K).
2. Find the most similar pair of documents in the document set.
3. Out of the documents of the selected pair, remove the one with the highest sum of similarities to other documents.
4. If the number of remaining documents equals K then go to step 5, else go to step 2.
5. The remaining documents will be used as initial cluster centres.

The simplest implementation can be based on the similarity matrix of documents. In our experiments reduction of similarities provided a significant improvement in the efficiency of clusters initialisation process (even though the $N - K$ steps need to be performed – for better efficiency, a random sample of data could be used). It may also cause that the outliers will be selected as seeds, so much better results are obtained with combination with the K-Medoids algorithm.

The best result for flat clustering methods in general was obtained with K-Medoids++ algorithm (see Sec. 6), it has to be, however, restarted a number of times to achieve good results and is non-deterministic.

4.3 Cluster Cardinality

Many clustering algorithms require the *a priori* specification of the number of clusters K. Several algorithms and techniques have been created to determine the optimal value of K automatically (Milligan and Cooper, 1985; Likas et al., 2001; Feng and Hamerly, 2007).

For K-means, we can use a heuristic method for choosing K according to the objective function. Define $RSS_{\min}(K)$ as the minimal RSS (see eq. 1) of all clusterings with the K clusters, which can be estimated by applying reduction of similarities technique. The point at which $RSS_{\min}(K)$ graph flattens may indicate the optimal value of K.

If we can make the assumption that $RSS_{\min}$ values are obtained through the RS, we can find the flattenings very fast and quite accurate. Moreover, the deterministic feature of the introduced method favours this assumption. Starting from the calculations of the $RSS_{\min}$ value for the largest K, we do not have to run the RS technique anew in the each next step. For $K - 1$ it is sufficient to remove only one centroid from K previously selected, thus the significant increase in performance is achieved.

4.4 Evaluation Method

Purity measure is a simple external evaluation method derived from information retrieval (Manning et al., 2009). In order to compute purity, in each cluster the number of documents assigned to the most frequent class in the given cluster is calculated, and then sum of these counts is divided by the number of all documents (N):

$$purity(C, L) = \frac{1}{N} \sum_{k} \max_{j} \mid C_k \cap L_j \mid \quad (2)$$

where $L = \{L_1, \ldots, L_m\}$ is the set of the (expected) classes.

The main limitation of purity is that it gives poor results when the number of clusters is different from the real number of valid classes. Purity can give irrelevant values if the classes significantly differ in size and larger ones are divided into smaller clusters. This is because the same class may be recognized as the most numerous in the two or more clusters.

We propose a simple modification to purity that helps to avoid such situations: let each cluster be assigned to the class which is the most frequent in a given cluster and only if this class is the most numerous in this cluster among all the clusters.

Hence, each class is counted only once (for a cluster in which it occurs most frequently) and

some of the clusters can be assigned to no class. This method of evaluation will be called **strict purity measure**. The value of strict purity is less than or equal to the standard purity calculated on the same partition.

5 Experiment Settings

To measure the distance between two documents d_i and d_j we used the cosine similarity defined as the cosine of the angle between their vectors:

$$sim_{\cos}(d_i, d_j) = \frac{\vec{d_i} \cdot \vec{d_j}}{\|\vec{d_i}\|\|\vec{d_j}\|} \quad (3)$$

The standard tf-idf weighting scheme was used.

Other types of distance measures and weighting models were considered as well, but preliminary tests showed that this setting is sufficient.

Table 1 presents text normalisations used in the experiment. Natural initial normalisations are `d,ch,m`, i.e.: (1) lowercasing all words (this ensures proper recognition of the words at the beginning of a sentence), (2) spell checking and (3) stemming using Morfologik package. All subsequent normalizations mentioned in this paper will be preceded by this initial sequence.

Symbol	Explanation
`ss`	Cut words after the sixth character
`m`	Stemming with Morfologik package
`ch`	Spell checking with Hunspell
`d`	Lowercase all words
`rs`	Remove only simple stop words
`rl`	Remove genre specific words
`rc`	Remove Polish city names
`p`	Remove stop words with abstract verbs, unwanted adverbs and Internet slang words
`t`	Use thesaurus to normalise synonyms, diminutives and augmentatives

Table 1: Text normalisations used.

6 Results

We compared a number of seeds selection techniques for the flat clustering algorithms using the smaller sub-corpus of 83 urban-legend texts. The results suggested that the K-Medoids++ and K-Medoids combined with centres selection by reduction of similarities perform better than other methods (see Table 2). The algorithms that are non-deterministic were run five times and the maximum values were taken. The best result for the development sub-corpus was obtained using Average Linking with the normalisation without any thesaurus.

Alg.	Purity	$\mathbf{P_{strict}}$	Purity	$\mathbf{P_{strict}}$
	`rs,t`		`p,rl,t`	
KM	0.783	0.675	0.762	0.711
KMd	0.831	0.759	0.871	0.847
KMpp	0.916	0.904	0.952	0.94
KMdpp	**0.988**	**0.976**	**0.988**	**0.976**
KM_{RS}	0.807	0.759	0.964	0.94
KMd_{RS}	0.965	0.918	**0.988**	**0.976**

Table 2: Comparison of flat clustering algorithms using the development set.

Results obtained for the test corpus are presented in Table 3. All tests were performed for the natural number of clusters ($K = 62$). Hierarchical methods proved to be more effective than flat clustering algorithms probably because the former do not seek equal-sized clusters. The best strict purity value (0.825) was achieved for the Average Linkage algorithm with the `p,rl,t,ss` normalisation. Average Linking was generally better than the other methods and gave results above 0.8 for the simplest text normalisation as well.

For the best result obtained, 22 clusters (35.5%) were correct and 5 clusters (8%) contained one text of an incorrect story type or did not contain one relevant text. For 10 classes (16%) two similar story types were merged into one cluster (e.g. two stories about dead pets: *the dead pet in the package* and *"undead" dead pet*). Only one story type was divided into two "pure" clusters (shorter versions of a legend were categorised into a separate group). The worst case was the *semen in fast food* story type, for which 31 texts were divided into 5 different clusters. A number of singleton clusters with outliers was also formed.

As far as flat algorithms are concerned, K-Medoids++ gave better results, close to the best results obtained with Average Linkage[11]. The

[11]The decrease in the clusters quality after adding some normalisation to K-Medoids++ algorithm, does not necessar-

value of 0.821 was, however, obtained with different normalisations including simple stemming. K-Medoids with reduction of similarities gave worse results but was about four times faster than K-Medoids++.

Both flat and hierarchical algorithms did not manage to handle the correct detection of small classes, although it seems that words unique to each of them could be identified. For example, often merged story types about student exams: *a pimp* (11 texts) and *four students* (4 texts) contain words *student* (= *student*), *profesor* (= *professor*) and *egzamin* (= *exam*), but only the former contains words *alfons* (= *pimp*), *dwója* (= *failing mark*), whereas *samochód* (= *car*), *koło* (= *wheel*) and *jutro* (= *tomorrow*) occur only in the latter. Similarly, topics *fishmac* (7) and *have you ever caught all the fish?* (3) contain word *ryba* (= *fish*) but the first one is about McDonald's burgers and the second one is a police joke. In addition, texts of both story types are very short.

Hierarchical methods produced more singleton clusters (including incorrect clusters), though K-Medoids can also detect true singleton classes as *in a willy* and *dad peeing into a sink*. These short legends consist mainly of a dialogue and seem to be dissimilar to others, so they have often been taken as initial centroids in K-Medoids$_{RS}$ and K-Medoids++. Topics containing texts of similar length are handled better, even if they are very numerous, e.g. *what is Your name?* (32). But this legend is very simple and has few variants. On the other hand, the popular legend *semen in fast food* (31) has many variants (as semen is allegedly found in a milkshake, kebab, hamburger, salad etc.).

The results confirm the validity of the proposed text normalisation techniques: better clusters are obtained after removing the non-standard types of words and with a thesaurus including diminutives and augmentatives. Further development of the thesaurus may lead to the increase of the clusters quality.

6.1 Guessing Cluster Cardinality

Fig. 1 presents the estimated minimal average sum of squares as a function of the number of clusters K for K-Medoids with centres selection by the RS

ily imply the worse effectiveness, but may suggest an unfortunate random selection of the first centroid.

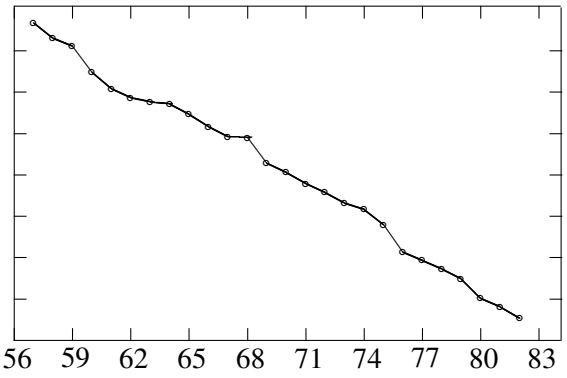

Figure 1: Sum of squares as a function of the K value in K-Medoids with seeds selection by reduction of similarities. Used normalization: `p,rl,t,td,ss`.

(i.e. minimal values of the $RSS(K)$ for each K is approximated with this technique). The most probably natural number of clusters is 67, which is not much larger than the correct number (62), and the next ones are 76 and 69. It comes as no surprise as for $K = 62$ many classes were incorrectly merged (rather than divided into smaller ones). The most probable number of guessed cluster cardinality would not change a wider range of K than one presented in Fig. 1 if were considered.

7 Conclusions

The clustering of urban-legend texts should be considered harder than e.g. clustering of news articles:

- An urban-legend text of the same story type may take very different forms, sometimes the story is summarised in just one sentence, sometimes it is a detailed retelling.
- Other legends are sometimes alluded to in a text of a given story type.
- The frequency of named entities in urban-legend texts is rather low. City names are sometimes used but taking them into account does not help much, if any (legends are rarely tied to a specific place or city, they usually "happen" where the story-teller lives). Hence it is not possible to base the clustering on named entities like in case of the news clustering (Toda and Kataoka, 2005).

Normalization	Words	KMd	KMd$_{RS}$	KMdpp	CmpL	AvL	WAvL
rs	7630	0.675	0.732	0.771	0.747	0.806	0.798
rs,rl	7583	0.694	0.742	0.776	0.772	0.789	0.776
rs,t	6259	0.699	0.743	0.805	0.779	0.799	0.766
rs,rl,t	6237	0.688	0.731	0.775	0.818	0.785	0.770
p,rl	7175	0.698	0.743	0.773	0.77	0.78	0.798
p,rl,rc	7133	0.697	0.739	0.794	0.773	0.758	0.795
p,rl,ss	6220	0.684	0.763	0.758	0.750	0.808	0.792
p,rl,t	5992	0.699	0.732	0.786*	0.806	**0.825**	0.778
p,rl,t,rc	5957	0.618	0.731	0.777	0.811	0.824	0.776
p,rl,t,ss	5366	0.719	0.775	**0.821***	0.789	0.813*	0.791
p,rl,t,rc,ss	5340	0.71	0.786	0.775	0.789	0.811	0.791
Mean	—	0.689	0.747	0.783	0.782	0.8	0.785

Table 3: Results of clustering urban-legend texts (strict purity) for algorithms: K-Medoids (KMd), K-Medoids++ (KMdpp), K-Medoids with seeds selection (KMd$_{RS}$), Complete Linkage (CmpL), Average Linkage (AvL) and Weighted Average Linkage (WAvL). Values with the star sign were obtained with the probabilistic document frequency instead of the idf.

- Some story types include the same motif, e.g. texts of distinct story types used the same motif of laughing paramedics dropping a trolley with a patient.
- Urban legends as texts extracted from the Internet contain a large number of typographical and spelling errors.

Similar problems will be encountered when building a system for discovering new urban-legend texts and story types.

Acknowledgement

The paper is based on research funded by the Polish Ministry of Science and Higher Education (Grant No N N516 480540).

References

David Arthur and Sergei Vassilvitskii. 2007. *k-means++: The advantages of careful seeding*. SODA 2007: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 1027–1035.

Pavel Berkhin. 2002. *Survey Of Clustering Data Mining Techniques*. Accrue Software, San Jose, CA.

Jan H. Brunvand. 1981. *The vanishing hitchhiker: American urban legends and their meanings*, Norton.

Jan H. Brunvand. 2002. *Encyclopedia of Urban Legends*, Norton.

Yu Feng and Greg Hamerly. 2007. *PG-means: learning the number of clusters in data*. Advances in Neural Information Processing Systems 19, 393–400, MIT Press.

Filip Graliński. 2009 *Legendy miejskie w internecie*. Mity współczesne. Socjologiczna analiza współczesnej rzeczywistości kulturowej, 175–186, Wydawnictwo Akademii Techniczno-Humanistycznej w Bielsku-Białej.

Anil K. Jain, M. Narasimha Murty and Patrick J. Flynn. 1999. *Data Clustering: A Review*. ACM Computing Survey, 31, 264–323.

Aristidis Likas, Nikos Vlassis and Jakob J. Verbeek. 2001. *The Global K-Means Clustering Algorithm*. Pattern Recognition, 36, 451–461.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

Glenn W. Milligan and Martha C. Cooper. 1985. *An examination of procedures for determining the number of clusters in a data set*. Psychometrika.

Anna D. Peterson, Arka P. Ghosh and Ranjan Maitra. 2010. *A systematic evaluation of different methods for initializing the K-means clustering algorithm*. Transactions on Knowledge and Data Engineering.

Martin F. Porter. 1980. *An algorithm for suffix stripping*. Program, 3, 130–137, Morgan Kaufmann Publishers Inc.14.

Hiroyuki Toda and Ryoji Kataoka. 2005. *A clustering method for news articles retrieval system*. Special interest tracks and posters of the 14th international conference on WWW, ACM, 988–989.