Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

Influence of *a priori* Knowledge on Medical Document Categorization

Łukasz Itert, Włodzisław Duch, *Senior Member, IEEE*, and John Pestian

***Abstract*—A significant part of medical data remains stored as unstructured texts. Semantic search requires introduction of markup tags. Medical concepts discovered in hospital discharge summaries are used to create several feature spaces. Experts use their background knowledge to categorize new documents, and knowing category of the document disambiguate words and acronyms. A model of document similarity to reference sources that captures some intuitions of an expert is introduced. Parameters of the model are evaluated using linear programming techniques. This approach is applied to categorization of the medical discharge summaries providing simpler and more accurate model than alternative text categorization approaches.**

I. Introduction

Automatic tools for conversion of unstructured medical texts into semantically-tagged documents are urgently needed because in medical domain errors may be dangerous and costly, medical vocabularies are huge and a very large number of abbreviations and acronyms are used. Critical differences between General English and Medical English have been analyzed in a numbers of publications [1]. The Cincinnati Children's Hospital Medical Center (CCHMC), a large pediatric academic medical center with over 700,000 pediatric patient encounters per year, has terabytes of medical data, mostly in form of raw texts, stored in a complex, relational database integrating many electronic hospital services [2].

Our long-term goal is to create tools for automatic annotation of unstructured medical texts, adding full information about all medical concepts, expanding acronyms and abbreviations and disambiguating all terms. Processing of medical texts requires three steps: 1) mapping strings of symbols to unique terms; 2) resolving ambiguities and mapping terms to concepts in the Unified Medical Language System (UMLS) Metathesaurus [3] that includes Semantic Network as one of the three UMLS Knowledge Sources, providing ontological relations for the Metathesaurus concepts; and 3) creating a full semantic representation of the text, that facilitates understanding and answering questions about its content. These 3 steps are all intimately connected and require the use of recognition, semantic and episodic memory [4].

Understanding texts is based on a prior knowledge that generates expectations of a few selected concepts, and inhibition of many others, a process that statistical methods of natural language processing [5] approximates in a very crude way, because co-occurrence relations are only a poor reflection of structured knowledge stored in human memory. Medical expert reading text quickly forms a hypothesis about the particular subdomain the text may be assigned to, and interprets the text in the light of this knowledge, as well as the background knowledge derived from medical studies, textbooks and individual experience. This is especially true if relatively short texts are analyzed, such as patient's Discharge Summaries, containing brief medical history, current symptoms, diagnosis, treatment, medications, therapeutic response and outcome of hospitalization, are analyzed. Many medical concepts appear very rarely in Discharge Summaries. Word Sense Disambiguation (WSD) or document categorization algorithms that are based only on the context relations but ignore the background medical knowledge are not useful here. Computational intelligence (CI) models based on cognitive inspirations should be quite useful in natural language processing and text mining problems. Although fuzzy techniques could be used for analysis of texts very large number of rules will be generated, therefore an alternative approach is proposed here.

The first step towards semantic annotation and disambiguation of medical Discharge Summaries requires discovery of the document topic. In the simplest case topic is the main disease that has been treated; automatic assignment of billing codes requires more detailed topics. Categorization of such documents requires a method of evaluation of their similarities, but two documents from the same category may contain very few common concepts. In the next section ways of defining useful feature spaces for medical documents are discussed. Similarity measures that take into account a priori knowledge are then introduced and a model trying to capture expert intuition is discussed. Estimation of parameters of this model is done using linear programming techniques. Numerical experiments with over 4500 discharge summaries

Lukasz Itert is with the Division of Rheumatology, 3333 Burnet Avenue, Children's Hospital Research Foundation, Cincinnati, OH 45229, USA, and with the Department of Informatics, Nicolaus Copernicus University, Toruń, Poland; email: Lukasz.Itert@cchmc.org

Wlodzislaw Duch is with the Department of Informatics, Nicolaus Copernicus University, Grudziądzka 5, 87-100 Toruń, Poland and with the School of Computer Engineering, Nanyang Technological University, 639798 Singapore. Contact info: Google: Duch

John Pestian is with the Department of Biomedical Informatics, Children's Hospital Research Foundation, Cincinnati, OH, USA; email: John.Pestian@cchmc.org

allow for comparison of this approach with traditional document categorization approaches.

The use of prior knowledge, including structured knowledge, in conjunction with classification of biomedical texts, is still quite rare, and mostly limited to feature selection level rather than representation of document topics [6]. We are not aware of any other attempts to use knowledge extracted from free texts in biomedical document categorization.

II FROM DOCUMENTS TO FEATURE SPACES

Medical texts differ from texts in general domain: they are usually not understood by the lay people, they are full of medical terms specific to a particular branch of medicine, frequently may violate grammar. Clinical texts are dictated or manually written, and thus contain frequent misspelling and typing errors, punctuation errors, large number of abbreviations and acronyms. Therefore the bag-of-words representation of such documents leads to very large feature spaces, many strongly correlated features (terms forming concepts), and extremely sparse representation. Unified Medical Language System (UMLS) [3] developed by the US National Library of Medicine is a huge ontology of medical concepts that may be used to discover useful concepts.

Discharge summaries have been retrieved from the hospital database using customized SQL queries, taking into account privacy issues, disease synonyms, duplicate records and so forth. All queries were executed across two different tables describing patient's discharge records and diagnosis. In this way a labeled set of documents has been created. Since patients were diagnosed with different diseases (which implies different symptoms, treatments, medications, etc.) each class should be characterized by unique and distinct vocabulary. This information may be used to find simple rules classification and in processing of discharge summaries for which diagnosis is missing.

Table I. Names of the diseases used in the study

Disease name	No. of records	Average size in bytes
Pneumonia	609	1451
Asthma	865	1282
Epilepsy	638	1598
Anemia	544	2849
Urinary tract infection (UTI)	298	1587
Juvenile Rheumatoid Arthritis	41	1816
Cystic fibrosis	283	1790
Cerebral palsy	177	1597
Otitis media	493	1420
Gastroenteritis	586	1375

For experiments reported here documents that belong to 10 distinct disease classes were selected. Except for the Juvenile Rheumatoid Arthritis (JRA) class that contained only 41 documents all the other classes were among the most common in the database containing discharge records. Information on the dataset created for these experiments, class distribution and the average length of documents in each class, is presented in Table I. Overall there are 4534 records with patients discharge summaries, with "asthma" being the majority class that covers 19.1%, defining the baseline for classification and outnumbering the smallest class about 15 times. All documents are short, less than 3000 characters, with the average length below 200 characters. With this type of class disproportions purely statistical approaches are bound to fail and the need for a priori knowledge is quite clear. Many rare diseases in out database have only very few documents.

The name of the disease that should be used as the category label plays a dual role: it is one of the features used to describe the document, and it is also the class label. For example, documents from the "asthma" class frequently contain the name "asthma", but they may also contain the names of other diseases. The frequency of appearance of each of the 10 disease names in the documents may be taken as the class indicator, giving a more informed base rate distribution. Using this approach leaves 55.3% of documents unclassified (including impasses, i.e. cases with several identical highest frequencies), 34.6% correctly classified and 10.1% errors.

Construction of the feature space is of primary importance here. Four spaces have been considered here: based on reference knowledge or on the analyzed documents (native space); in each case individual words or semantic concepts may be used. The for spaces are called word-reference, semantic-reference, word-native and semantic-native.

A. *Data Preprocessing*

At this stage each text is subject to several processing techniques: exhaustive set of parsing rules are used to handle all punctuation issues, numbers, special characters (#,@,!) and internal separators that should be removed.

Typical methods for text dimensionality reduction include stemming to find unique form of words [7], and stop-word list of common English words to remove words that do not contribute to document categorization [8]. Several versions of a word-based space may be taken into account, depending on the procedures used. Here only the word space reduced with stop words and then stemmed is considered.

Medical concept discovery maps fragments of texts on medical ontologies, finding unique names, introducing features that are more specific, capturing some semantics of the medical texts, and avoiding strongly correlated features (parts of multi-word concepts). MetaMap Transfer (MMTx) program package is a collection of lexical and semantic tools, designed and optimized for exploring biomedical text resources [9]; as a result text is annotated using UMLS Metathesaurus concepts. The UMLS system is a big source of concepts with many interpretations and meanings [3]. Unfortunately this mapping is usually not unique and suffers from word ambiguity and other problems. To avoid false-positive mappings a very restrictive MMTx settings has been used during string matching. Concepts are assigned to a number of semantic types and in document categorization it is advantageous to focus attention on specific types (such as

Antibiotics or Syndrome), ignoring more general types (such as Temporal or Qualitative Concepts).

Latent Semantic Indexing (LSI) [10] is a well-known unsupervised technique for feature space dimensionality reduction. New features are found by taking the principal components (usually using Singular Value Decomposition), or weighted combinations of original features, corresponding to the highest variance in the document space. In medical document categorization a single specific occurrence of a concept may be an important indicator of the document category, while the contribution of this concept to the principal components will be negligible. Features created by linear or non-linear combinations loose their semantics, while concept discovery enhances it.

B. Creation of the Feature Spaces

Two collections of texts are used: reference to capture *a priori* knowledge, and clinical, based on summary discharges. For each collection two types of feature spaces have been created: word and semantic. In both cases feature values represent frequency of words or frequency of concepts, respectively. Since categorization is done only for the clinical data features space created from this data is referred to as "native", while the reference space is based on external (non-native) source of data. Since it's quite likely that reference text will contain words/concepts not found in the clinical data, after creating a reference space it is automatically narrowed down to match the existing clinical words/concepts. The initial number of words that are good candidates for features in the reference space was 4008, the space has been finally limited to less than 2299 features using the stemming algorithm, stop-list and removing the features that appeared only once in all reference texts. Exactly the same text parsing algorithms applied to the collection of clinical data has been used to create word-native space. The number of candidate words is 30260, and even after the same reduction as used for the reference space still 13248 words remain (many proper names, spelling errors, alternative spellings, abbreviations, acronyms, etc).

Creation of semantic feature spaces requires more sophisticated processing methods. Medical records contain many specific, unique and uncommon words, therefore extremely large feature space may easily be created. To reduce it key concepts in the collection of documents are identified and grouped by their semantic types. The ULMS Metathesaurus – a collection of lexical and semantic information about biomedical concepts, their various names, and relationships among them – includes more than one million concepts represented by more than 4 million strings. Each concept is assigned to one of the 135 semantic types. Only 26 of these semantic types were used (listed in the first column of Tab. III) because they are specific, medical concepts useful in document categorization. Thus using the UMLS ontology as a base all common words may be filtered out and non-medical concepts excluded.

At first, the native semantic feature space has been created. Each of the 4534 documents has been processed by the MMTx software [9] and the key concepts have been filtered, leaving only those that belong to one of the predefined semantic types. The final number of features included in the native space based on concepts discovered in medical records was 7220 (this yields 7040 features since some of the UMLS concepts are assigned to more than one semantic type). These concepts appeared in medical records 195321 times (Tab. III, column I-All).

Next, reference documents describing each of the 10 diseases (Tab. II, see more on these texts below) have been preprocessed, and medical concepts discovered using the MetaMap software and restricting the semantic types to those most specific from the medical point of view (Tab. III, column II). As a result 1097 unique concepts have been identified, appearing 4436 times in the reference texts. Each prototype of the disease may be represented in this feature space as a vector with components obtained from scaled frequencies of a given concept in the reference document describing a given disease. This forms reference feature space with reference vectors that represent background knowledge for specific diseases. For each of the features (dimensions) in this space at least one reference vector has the corresponding component with non-zero value. This is not true for medical records, where only a subset of 807 unique features have non-zero components and thus as long as experiments are restricted to the selected texts there is no need to use the remaining 290 features (Tab. III, column III). Similarity between clinical and reference records may be computed in this space. Background knowledge contained in features that appear only in the reference space, but not in the limited selection of medical records taken for analysis, should be useful if new texts will be retrieved from the database.

Some features that are specific to a particular disease should be given more weight. This may be done using references vectors that contain weights; for example, if only one disease mentions HIV virus the corresponding entry in the reference vector should be given a high weight. Scaling factors of this sort may be introduced at the later stage using "term weighting" or optimization of similarity functions, for example based on conditional probabilities.

Table II. Information about reference documents.

Disease name	Size (bytes)
Pneumonia	23583
Asthma	36720
Epilepsy	19418
Anemia	14282
Urinary tract infection (UTI)	13430
Juvenile Rheumatoid Arthritis (JRA)	27024
Cystic fibrosis	7958
Cerebral palsy	35348
Otitis media	32416
Gastroenteritis	9906

Filtered reference space is relatively small, with 807 features that correspond to important medical concepts discovered in the reference texts. The texts given for analysis contain many more unique concepts than found in the reference texts, in our case 7220. For example, only one vitamin has been men-

tioned in the reference texts, while 30 vitamins have been mentioned in the medical records (Tab. III). Nevertheless, the filtered semantic-reference space with only 807 dimensions (11.2 % of native features) covers over 80 thousands concepts which account for 41.3 % of all concepts found in clinical data (Tab. III, the very last column).

C. Reference Data

The information contained in short texts, such as the summary discharges analyzed here, may by itself not be sufficient for proper categorization. An expert reading such texts brings into this process rich background knowledge derived from textbooks and medical practice. This knowledge is partially contained in disease definitions found in online dictionaries, textbooks and ontologies. A typical disease definition consists of the following sections: definition, cause; incidence, risk factors, symptoms, signs and tests, treatment, expectations (prognosis), complications. Texts taken from medical books are quite long and may contain many concepts that refer to rare situations that will not be commonly encountered, but still form a are very important part of expert's knowledge.

This reference knowledge may be represented in many ways. For classifiers based on similarity or requiring numerical representation in a vector space concept-based feature space is quite appropriate. Each reference disease may be represented by a single vector, or by a number of vectors, with frequencies of concepts that may be expected, estimated using the reference knowledge sources. A single vector represents a general prototype of a disease, but most diseases have several variants, with slightly different combination of symptoms. For example, the text describing a given disease may mention that at least 3 of the 5 symptoms listed in the text should appear, therefore in the 5-dimensional subspace the reference vectors may contain non-zero frequencies for all 5 symptoms (1 vector), 4 out of 5 (4 vectors), or 3 out of 5 (10 vectors), requiring altogether 15 vectors to represent all combinations. If there are several such groups of alternative concepts (causes, symptoms, treatments, complications etc) the number of references vectors will grow in a combinatorial way. Creation of such reference vectors may require deeper understanding of medical texts, and thus will be rather difficult to automatize.

Naive approach to background knowledge includes all concepts that appear in reference knowledge sources for a given disease, resulting in a single prototype vector that is rather different from any real case, where only a subset of symptoms or treatments appear. This is quite evident if documents from a given class are compared with the reference vectors created in this way. Proper scaling of similarity between reference vectors and document vectors may alleviate this situation, therefore this method will be used in combination with similarity-based methods as the most straightforward approach to incorporation of *a priori* knowledge into a similarity-based classifier system.

The reference texts were taken from MedicineNet [11], Children's Hospital Boston. Child Health A to Z [12], and the MedlinePlus: Medical Encyclopedia [13]. The size of these texts for each of the diseases is presented in Tab. II. More detailed description of the disease could result in a larger number of useful features.

II. Model of Similarity

In principle vector representation of documents is not necessary if a method that could estimate similarity $S_{ij}=S(D_i,D_j)$ directly from text comparison could be devised. In practice direct evaluation of similarity is not possible and numerical representations based on term frequency are used as the starting point. In the most common approach documents D_i of length $l_j=|D_i|$ are composed of terms (words or collocations).

A. Term weighting

Term frequencies *tf* for term $i = 1 \ldots n$ in document D_j are calculated for all documents that should be compared. Term frequencies are then transformed to obtain features such that in the feature space simple metric relations between vectors representing documents should reflect their similarity. This transformation should avoid giving too much weight to features that appear with high frequency, and to long documents that tend to have more non-zero frequencies and higher frequencies. There are many ad-hoc ways to introduce such weights. For example, for non-zero term frequencies [5]:

$$s_{ij} = round\left(10\times\frac{1+\log tf_{ij}}{1+\log l_j}\right) \qquad (1)$$

Words that appear in all documents may have high frequency, but carry little information that could be used for document categorization. Uniqueness of each feature is inversely proportional to the number of documents this feature appears in; if the term i appears in df_i out of N documents the final weighting is:

$$s_{ij} = round\left(10\times\frac{1+\log tf_{ij}}{1+\log l_j}\log\frac{N}{df_i}\right) \qquad (2)$$

This is usually called *tf* x *idf* weighting scheme. If the term i appears in all documents it does not contribute and $s_{ij}=0$ for all j. The *tf* x *idf* weighting scheme may take some variants. For example, in the Smart system [14] term frequencies are rescaled to [0.5,1], using $s = 0.5(1 + tf/maxtf)$, and in the Inquery system by $s=0.4+0.6\,tf/maxtf$) [15], but these weightings favor long documents. To avoid it normalization of the *tf* x *idf* scaled vectors is used as the final feature vectors X_j, that is all vectors $(s_{ij},\ldots s_{n,j})$ are divided by their length to obtain X_j.

$$s_{ij} = round\left(10\times\frac{1+\log tf_{ij}}{1+\log l_j}\log\frac{N}{df_i}\right);\quad X_j=\frac{s_j}{\|s_j\|} \qquad (3)$$

This normalization tends to favor shorter documents. More sophisticated normalization method has been introduced in the information retrieval to counter this effect [5][14], but unbiased normalizations have not yet been found. In document categorization we are interested in distribution of a given term among different categories, there-

Table III. The dimensionality of different spaces ("unique" column) and the total number of such concepts found in data ("all" column). I - semantic-native, II - semantic-reference, III - filtered semantic-reference (final space). Last column presents the number of concepts found in the real data using space III.

Semantic type	I		II		III		Data III
	Unique	All	Unique	All	Unique	All	All
Anatomical Structure	20	186	4	13	3	11	116
Antibiotic	100	7664	16	95	16	95	3096
Bacterium	98	1850	13	69	9	65	627
Biologically Active Substance	148	6908	24	80	15	64	2052
Biomedical or Dental Material	53	1192	5	8	5	8	57
Body Location or Region	196	5298	18	93	17	91	3638
Body Part, Organ, or Organ Component	633	8777	113	558	87	511	3879
Body Space or Junction	84	478	4	81	3	51	49
Body Substance	75	8881	27	152	21	145	2872
Body System	20	907	10	71	6	55	166
Clinical Attribute	63	840	8	23	7	21	244
Clinical Drug	88	271	2	2	0	0	0
Diagnostic Procedure	236	10599	47	126	35	108	6870
Disease or Syndrome	1378	20132	248	1415	174	1293	12027
Enzyme	74	1928	6	16	5	15	504
Finding	1094	29770	126	325	89	283	7212
Hormone	60	1891	7	54	6	52	998
Laboratory or Test Result	143	1824	13	36	7	23	581
Laboratory Procedure	250	8113	41	86	29	58	3179
Organ or Tissue Function	108	3542	27	61	17	36	734
Pharmacologic Substance	903	24214	134	278	90	212	6030
Physiologic Function	40	4273	11	76	8	73	3041
Sign or Symptom	573	22518	116	522	99	500	15621
Therapeutic or Preventive Procedure	736	22254	68	148	53	129	6829
Virus	17	485	8	47	5	42	260
Vitamin	30	526	1	1	1	1	20
Total	7220	195321	1097	4436	807	3942	80702

fore instead of the $\log(N/df_i)$ factor the logarithm of ratio $\log(K/cf_i)$ of the number of classes K to the number of classes cf_i term i may be found is used.

B. Evaluation of document similarity

Euclidean or other simple distance measures do not capture intuitive estimation of document similarity. Prior to the examination of a document the probability that it belongs to category C should be equal to the *a priori* probability $p(C)$. The background knowledge of an expert about the reference documents from class C_i may be represented using term frequencies $R_i(x_j)$ for the term x_j. These frequencies are collected in the reference vector R_i.

Proposition 1: the initial distance for an unknown document D to the reference vectors R_i should be proportional to $d_{oi}=|D-R_i|\sim 1/p(C_i)-1$. For rare classes this distance is large, for a very probable class $p(C_i)\approx 1$ it approaches 0. If the document does not contain any useful information it is close to the majority class. If term x_j has zero frequency in reference documents as well as in the document D the distance $d(D,R_i)$ is not changed from its current value (initially d_{oi}).

Proposition 2: if the term x_j appears in R_i with frequency $R_i(x_j)$ but does not appear in D the distance $d(D,R_i)$ should increase by $\Delta_i(x_j) = aR_i(x_j)$, where a is an adaptive constant. If a term appears frequently in reference documents from class C_i but does not appear in the document our belief that the document D is of the class C_i should decrease, thus the distance should increase.

Proposition 3: if a term x_j does not appear in R_i but it has non-zero frequency $D(x_j)$ in the document the distance $d(D,R_i)$ should increase by $\Delta_i(x_j)=bD(x_j)$.

Proposition 4: if a term x_j appears in both vectors and frequency $R_i(x_j)>D(x_j)>0$ the distance $d(D,R_i)$ should decrease by $\Delta_i(x_j)=-cD(x_j)$.

Proposition 5: if a term x_j appears in both vectors and frequency $D(x_j)>R_i(x_j)>0$ the distance $d(D,R_i)$ should decrease by $\Delta_i(x_j)=-eR_i(x_j)$.

Other contributions to distance could be considered but these propositions seem t capture some intuitive properties of

document similarity; if a term appears in both vectors than the distance is decreased by a constant times the smaller term frequency. For small term frequencies this situation may happen by pure chance, therefore smaller of the two frequencies is taken. If both frequencies are large this is a strong indication and should lead to a significant decrease of the current distance estimation.

Additional measure of term specificity is given by class-conditional probability $p(x_j|C_i)$. If a given term appears only in documents from the C_i class obviously it should be more important than if it appears with small probability for all classes. What with the intuition about terms that never appear in some classes (negative correlation)? They should increase the distance, and they do in Propositions 2 and 3, although only for zero frequencies.

Proposition 6. The final probability that a document D belongs to class C_i, including the contribution of all terms x_j, should be proportional to:

$$S(C_i \mid D;R_i) = 1-\sigma\left(\lambda\left[d_{0i}+\sum_j p(x_j \mid C_i)\Delta_i(x_j)\right]\right) \quad (4)$$

Here $\Delta_i(x_j)$ depends on 4 non-negative adaptive parameters *a, b, c,* and *e* that may be specific for each class, and the distance depends on the d_{0i} that may also be treated as an adaptive parameter; the slope λ is an additional parameter, giving 6 adaptive parameters per class. These parameters may be similar in each class and thus instead of optimizing them independently for each class one set of 6 parameter may be found. Weighted term contributions may sum to a negative number, giving small values after filtering through $\sigma(\cdot)$ function.

Probabilities are estimated after normalization:

$$P\left(C_i \mid D;R_i\right) = \frac{S\left(C_i \mid D;R_i\right)}{\sum_k P\left(C_k \mid D;R_k\right)} \quad (5)$$

This approach seems to capture most human intuitions when texts are analyzed using background knowledge. Parameters *a–e* may be estimated jointly for all classes or separately for each class using linear programming techniques.

III. Linear programming

In the linear programming [16] optimization goal is formulated by the linear problem (LP) with constraints:

$\min\{C^T X:\ \mathbf{A}X \geq B;\ C,X \in R^n,\ B \in R^m\}$

where **A** is a *n* x *m* matrix, B and C are known vectors and X is a vector of variables to be estimated. The expression $C^T X$ is the objective function and inequalities $\mathbf{A}X>\mathbf{b}$ are called the constraints. In practice it may be impossible to satisfy all the constraints (such LP problems are called infeasible). One way to search for a "maximally feasible solution" is to introduce slack variables ζ to restore feasibility. Using slack variables the modified system of inequalities can be rewritten as:

$\min\{\ \zeta = C^T X:\ \mathbf{A}X = B,\ \zeta > 0;\ C,X \in R^n,\ B \in R^m\}$

LP problems are solvable in polynomial time using interior point based methods [17]. Another popular class of algorithms for LP is based on the simplex algorithm [16]. To calculate coefficients *a–e* in our similarity measures PCx algorithm with ζ parameters has been used [18]. The condition

$$\min\left\{d_{0i}+\sum_j p(x_j \mid C_i)\Delta_i(x_j)\right\} \quad (6)$$

maximizes Eq. 4, that is the similarity measure between documents and reference vectors. However there are many classes in data and obviously the right class should be promoted and the incorrect ones penalized. This can be achieved by adding several constraints (7):

$$d_{0k}+\sum_j p(x_j \mid C_k)\Delta_k(x_j) \leq d_{0i}+\sum_j p(x_j \mid C_i)\Delta_i(x_j)$$

where *k* indicates the desired class and $k \neq i$. For each single training vector $K-1$ constraints were created (K is number of classes). The above inequality can be rewritten in a typical constraint form (8):

$$\sum_j\left[p\left(x_j \mid C_k\right)\Delta_k\left(x_j\right)-p\left(x_j \mid C_i\right)\Delta_i\left(x_j\right)\right] \leq d_{0i}-d_{0k}$$

Two different cases have been considered: one set of parameters for all classes (I) and separate set for each class (II).

Case I. If parameters *a*, *b*, *c*, *e* are class independent a single constraint takes the form:

$$\alpha a+\beta b+\gamma c+\delta e \ \leq d_{0i}-d_{0k} \quad (9)$$

where α, β, γ and δ coefficients depend on $p(x_j|C_i)$ and Δ_i calculated according to the propositions 1-6. These conditions can be presented in a matrix form using A matrix with dimensionality $4\cdot(K-1)\cdot N$ by 4.

Case II. Parameters *a, b, c, e* are now *K* dimensional vectors and a single constraint has *K* times more components:

$$\alpha^T a+\beta^T b+\gamma^T c+\delta^T e > d_{0k}-d_{0i} \quad (10)$$

where $\alpha^T a = a_1\alpha_1 + a_2\alpha_2 + \ldots + a_K\alpha_K$ (11)
and the same goes for $\beta^T b$, $\gamma^T c$, $\delta^T e$.

Satisfying all K–1 inequalities (9) or (10) for one document *D* guarantees that its similarity measure (4) is maximal for the correct class. This provides the correct classification of document *D* (considered to be "clean"). Since each single constraint always links only two classes at a time, $4(K-2)$ parameters are always equal to 0 in these inequalities.

IV. Results of numerical experiments

The performance of different classifiers has been evaluated on different versions of transformed data including the most common and widely used text smoothing methods. Since the dimensionality of the problem is significant, the feature/class correlation has been computed and analyzed. Other results reported here include classification accuracies

with and without the reference vectors. Finally, results of the method based on linear programming optimization proposed here are reported. All calculations except the nearest-neighbor-based classification with prototypes were carried out using stratified 10-fold crossvalidation.

Experiments with feature ranking based on Pearson's linear correlation coefficients (CC) have been performed to estimate feature/class correlations. There are many weakly correlated features but in experiments with the kNN classifier using semantic space and one class against all other discrimination it was found that a CC threshold as small as 0.05 dramatically decreases accuracy, from over 95% on all features to below 50%. Similar results are obtained with various feature spaces and classifiers. Therefore even weakly correlated features cannot be disregarded in classification problems without significant loss of accuracy.

A. Results without the reference vectors

A few well known classification methods have been used to estimate the background or reference accuracy – the accuracy which could be obtained without additional reference knowledge (Table IV). Results are presented only for the *tf*-normalized semantic space, as other results are not better. As rule-based data understanding is quite important in this case two decision trees have been used: C 4.5 [19] and SSV Trees [20] as implemented in the Ghostminer package [21]. In addition k-nearest neighbor (kNN) and SVM methods have been used (also using the Ghostminer package) for comparison as the reference knowledge is presented in form of prototype vectors, one per class. kNN showed much better training results than decision trees, and also significantly better test results than decision trees, although overall these results are still rather poor.

Table IV. Classification accuracy (in %) using 10-CV for *tf*-normalized semantic feature space data.

	kNN	SSV	C4.5
Train	93.7	47.4	27.9
Test	48.9	39.5	34.9

Table V. Best 10-CV test accuracies across different data normalizations. M0: *tf*, M1:binarized, M2: $s_{ij} = \sqrt{tf}$, M3: s_{ij}= 1 + log(*tf*), M4: s_{ij}= (1+log(tf_{ij}))log(N/df_i), M5: Eq (3)

	M0	M1	M2	M3	M4	M5
kNN	48.9	50.2	51.0	51.4	49.5	49.5
SSV	39.5	40.6	31.0	39.5	39.5	42.3
SVM	59.3	60.4	60.9	60.5	59.8	60.0

B. Results with the reference vectors

The reference knowledge from medical textbooks has been presented in the form of prototype vectors, one per class. This leads to a greatly simplified nearest neighbor method, as distances to only 10 reference vectors have to be checked and the most similar vector selected. Two distance functions have been considered, Euclidean and cosine dissimilarity measure. All data becomes now the test data as the documents to be classified have not been used to create the reference model.

Independently of the feature space and data normalization methods the Euclidean distance leads to a very poor performance (6-15% of accuracy) since the majority of vectors are assigned to the class 7 (Cystic fibrosis) as the reference vector representing this class is the shortest (Table II). The simplest *tf* normalization gives over 60% accuracy with cosine distance, a significant improvement over kNN with much simpler model.

Table VI. Accuracies (in %) across different data normalizations using only reference vectors. M0-M5, as in Tab. V.

kNN	M0	M1	M2	M3	M4	M5
Euclidean	6.2	6.2	6.2	6.3	15.0	6.2
cosine	60.1	58.9	56.7	56.8	56.5	43.8

C. Optimized similarity function

The approach described in Sec. II and III has been used to optimize the coefficients *a*, *b*, *c*, *e*, that should capture intuitive evaluation of document similarity. Two cases have been studied, with common parameters for all classes, or separate parameters for each class. In this case linear programming on the training crossvalidation partition has been used to optimize these parameters and similarity to all K=10 reference vectors calculated to classify the test data. The β parameter for both results were set to 0.01. For higher values of β the great increase in the number of near impasses could be noticed since for number of vectors probabilities Eq. (5) for different classes were close to 1.

Case I: For each 10-CV step on average about 95% of all constraints were satisfied, however the number of vectors for which correct unique class could be assigned was much lower (~ 61%), giving classification accuracy of 61.1%

Case II: With 92% of satisfied constraints the number of "clean" vectors was approximately 10% higher than in the Case I. The final 10-CV accuracy reached 71.6%. This is quite significant improvement comparing to all other results on this data. In all calculations reported here variance of the test results was below 2%.

V. Conclusion

Full annotation of unstructured documents that may facilitate semantic analysis of texts is a great challenge. Assigning documents to specific categories that will assist in disambiguation of terms and concepts should be treated as the first step towards this goal. Medical texts are rather specific, containing very large number of unique concepts. Direct standard approach to the document classification, based on vector representation using the *tf* x *idf* weighting scheme leads to quite poor results using the nearest neighbor and decision trees approaches. This is primarily a deficiency of the naive

document representation, but also lack of *a priori* knowledge needed for categorization of these documents. Many useful tools were created to help in medical document analysis: spelling tools, large ontologies such as the UMLS [3] and software for mapping text to concepts. It is clear that knowledge contained in medical records, such as discharge summaries analyzed here, is by itself not sufficient to categorize them with high accuracy. Therefore reference texts that systematically describe each disease have been introduced. These texts were analyzed using the MetaMap software [9] to discover concepts that belong to one of the 26 useful semantic types describing specific medical entities.

The use of *a priori* knowledge in computational intelligence is an important topic that may be approached from different perspectives [22][23]. Fuzzy rule-based systems for text mining are quite difficult to create because the number of concepts that one has to consider is huge and thus the complexity of the whole system is going to be large. Recently we have shown that fuzzy rules may be derived directly from prototype-based rules [24]. Similarity-based methods may therefore be useful not only in predictive methods [25] but also in data understanding. In the discharge summary categorization highest accuracies were obtained using the nearest neighbor method, giving additional justification to focus on prototypes rather than fuzzy rules. The simplest knowledge representation, in form of a single reference vector per class, has been used, with the reference space build on the set of concepts derived from the reference texts for each disease. Some of these concepts never appear in our database of medical records, but may still be useful if new documents will be given for analysis. A new approach to the evaluation of similarity of documents that refers to the background knowledge and captures some human intuitions has been introduced. As a result a simple model with a few parameters optimized using linear programming has been created, giving surprisingly large increase of accuracy compared to the kNN, decision trees or SVM classifiers.

Finding the simplest decomposition of medical records into classes using either sets of logical rules or minimum number of prototypes, is an interesting challenge. Although much remains to be done before unstructured medical documents and general web documents will be fully and reliable annotated in an automatic way *a priori* knowledge certainly will be very important. Increasing the number of reference vectors in each class could be done if a detailed textbook description of all subtypes of a given disease was available. This is probably the knowledge that medical doctors gain through the years of practice and frequently it is never verbalized. In fact prototype-based approach may be treated as a crude approximation to the activity of neural cell assemblies in the brain of a medical expert who thinks about a particular disease. Creating better approximations to the representation and the use of knowledge in this process is a great challenge for CI. A fascinating possibility suggested by our results is to use data mining techniques to discover major subtypes of disease that could improve categorization of classifiers but also help in training of young medical doctors by presenting optimal sets of cases for their study. Textbook knowledge is frequently not sufficient (for example, there are many new drugs mentioned in our documents that have not been mentioned in the textbooks), and thus some reference vectors derived from clusterization of the actual data should also be added as prototypes. With sufficient amount of documents optimization of individual feature weights could also be attempted. Many other ideas are currently being pursued.

ACKNOWLEDGMENT

W. Duch thanks the Polish Committee for Scientific Research, research grant 2005-2007, for support.

REFERENCES

[1] D. Campbell, SB. Johnson, Comparing syntactic complexity in medical and non-medical corpora. Proc. of the AMIA Annual Symposium, pp. 90-95, 2001.
[2] J. Pestian, B. Aronow, K. Davis, Design and Data Collection in the Discovery System. Int. Conf. on Mathematics and Engineering Techniques in Medicine and Biological Science, 2002.
[3] UMLS Knowledge Sources, 13th Edition – January Release. Available: http://www.nlm.nih.gov/research/umls
[4] J.R. Anderson, Learning and Memory J. Wiley and Sons, NY 1995.
[5] C.D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing MIT Press, Cambridge, MA 1999.
[6] J. Mostafa,W. Lam, Automatic classification using supervised learning in a medical document filtering application. Information Processing and Management: an International Journal vol. 36, issue 3, pp. 415-444, 2000.
[7] M.F. Porter, An algorithm for suffix stripping, Program, 14(3): 130-137, 1980,
[8] Package Lingua::EN::StopWords from http://www.cpan.org
[9] MetaMap, available at http://mmtx.nlm.nih.gov
[10] S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman, "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407, 1990.
[11] Available at http://www.medicinenet.com
[12] Available at http://web1.tch.harvard.edu/cfapps/A2Z.cfm
[13] Available at http://www.nlm.nih.gov/medlineplus/encyclopedia.html
[14] A. Singhal, C. Buckley, M. Mitra, Pivoted Document Length Normalization. ACM SIGIR Conference, Zurich, 1996, pp. 21-29.
[15] J. Broglio, J. Callan and W.B. Croft, "Inquery system overview". Proc. of the TIPSTER Text Program (Phase I), pp. 47-67. San Francisco, CA: Morgan Kaufman Publishers Inc. 1994.
[16] R.J. Vanderbei, Linear Programming. Foundations and Extensions. 2nd ed, Springer 2001.
[17] S. Wright. Primal-Dual Interior-Point Methods. SIAM Publications, 1997
[18] J. Czyzyk, S. Mehrotra, M. Wagner and S. J. Wright: PCx: An Interior-Point Code for Linear Programming, Optimization Methods and Software vol. 12, pp. 397-430, 1999.
[19] J.R. Quinlan, C 4.5: Programs for machine learning. Morgan Kaufmann, San Mateo, CA, 1993.
[20] K. Grabczewski and W. Duch, The separability of split value criterion, 5th Conf. on Neural Networks and Soft Computing, Zakopane, Poland, 2000, pp. 201-208.
[21] Ghostminer data mining software, www.fqspl.com.pl/ghostminer/
[22] W. Pedrycz, Knowledge-Based Clustering: From Data to Information Granules, J. Wiley, N. York 2005.
[23] W. Pedrycz, Knowledge-Based Clustering in Computational Intelligence. In: W. Duch and J. Mańdziuk, Challenges for Computational Intelligence (2007, in print).
[24] W. Duch and M. Blachnik, Fuzzy rule-based systems derived from similarity to prototypes. Lecture Notes in Computer Science vol. 3316, pp. 912-917, 2004.
[25] W. Duch, Similarity based methods: a general framework for classification, approximation and association. Control and Cybernetics 29, pp. 937--968, 2000.