Constructing Information Networks from Text Documents

Matjaž Juršič[1], Nada Lavrač[1,2], Igor Mozetič[1], Vid Podpečan[1], Hannu Toivonen[3]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia
[3] Dept. of Compute Science, FI-00014 University of Helsinki, Finland
{matjaz.jursic, nada.lavrac, igor.mozetic, vid.podpecan}@ijs.si,
hannu.toivonen@cs.helsinki.fi

Abstract. A major challenge for next generation data mining systems is creative knowledge discovery from diverse and distributed data/knowledge sources. In this task, an important challenge is information fusion of diverse representations into a unique data/knowledge format. This paper focuses on the graph representation of data/knowledge generated from text documents available on the web. The problem addressed is how to efficiently and effectively create an information network, named a BisoNet, from large text corpora. Several options concerning node and arc representation are discussed, and a case study information network is created from articles concerning autism, downloaded from the PubMed repository of medical publications. Open issues and lessons learned concerning representation choices are discussed

1 Introduction

Information fusion can be defined as the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human and automated decision making [Bos07]. Creative knowledge discovery can only be performed on the basis of a sufficiently large and sufficiently diverse underlying corpus of information. The larger the corpus, the more likely it is to contain interesting, still unexplored relationships.

The diversity of data/knowledge sources demands a solution that is able to represent and process highly heterogeneous information in a uniform way. This means that unstructured, semi-structured and highly structured content needs to be integrated. Information fusion approaches are diverse, and domain dependent. For instance, recent investigations in using information fusion to support scientific decision making within bioinformatics include [Dur06, Rac05]. [Smi06] exploit the idea of formulating an ontology-based model of the problem to be solved by the user and interpreting it as a constraint satisfaction problem taking into account information from a dynamic environment.

In this paper, we explore a graph-theoretic approach [Alb02, Bal06] which appears to provide the best framework to accommodate the two dimensions of information source complexity – type diversity as well as volume size. Efficient management and processing of very large graph structures can be realized in suitable distributed computing environments, such as grids, peer-to-peer networks or service-oriented architectures on the basis of modern database management systems, such as XML, object-oriented or graph-oriented database management systems. The still unresolved challenge of graph-theoretic approaches is the creation, maintenance, and update of the graph elements in the case of very large and diverse data/knowledge sources.

This paper focuses on the creation of large graph representations of data/knowledge from text document resources available on the web. The problem addressed is how to efficiently and effectively create an information network, named a BisoNet, from large text corpora. A BisoNet representation, as investigated in the BISON[1] project and discussed in [Ber08] is a graph representation, consisting of labelled nodes and edges. The original idea underlying the BISON project was to have a node for every relevant concept of an application domain, captured by terms denoting these concepts, that is, by "named entities". For example, if the application domain is drug discovery, the relevant (named) entities are diseases, genes, proteins, hormones, chemical compounds etc. The nodes representing these entities are connected if there is evidence that they are related in some way. Reasons for connecting two terms/concepts can be linguistic, logical, causal, empirical, a conjecture by a human expert, a co-occurrence observed in documents dealing with the considered domain. E.g., an edge between two nodes may refer to a document (for example, a research paper) that connects the represented entities.

Open issues in BisoNet creation are how to identify entities and relationships in data, especially from unstructured data like text documents: i.e., which nodes should be created from text documents, what edges should be created, what are the attributes with which they are endowed and how should edge weights be computed. This paper discusses several possible choices that can be made concerning the entities that constitute nodes and edges in a graph when the target knowledge representation is a BisoNet.

Another core question is the granularity chosen for describing the network elements, as well as the diversity of resources. To illustrate a great variety of text sources we use two extreme examples. Firstly, there is a concept of a generic document. We usually do not know much about texts from these sources, sometimes we do not even know which topics they describe. A general document can also contain a lot of noise. Examples of general documents are: a random text from the internet, blogs, newsgroup posts, mobile messages (sms) or mail archives. On the other extreme there are documents from well defined sources. These documents share a predefined

[1] Bisociation Networks for Creative Information Discovery: http://www.BisoNet.eu/.

vocabulary, we precisely know the subject they describe, and usually they are annotated with keywords. Text of this kind is often written by experts in some area who use a similar language to describe similar concepts. Sometimes we can even get access to an ontology or a hierarchy of concepts used in the documents. Examples of these documents are scientific articles from various domains and other documents from well structured and controlled sources (e.g.: encyclopaedia articles).

In this paper we use the example from the second of the two extremes. As a representative of a set of scientific documents we used subsets of medical articles from the PubMed[2] database, in combination with MeSH[3], a controlled vocabulary hierarchical thesaurus. A case study information network is presented, created from articles concerning autism, downloaded from the PubMed repository of medical publications. The open issues concerning representation choices are discussed in substantial detail.

The paper is structured as follows: The second section provides the problem description and outlines the structure of the solution proposed in this paper. The next section sets the standard terminology used in the area of text mining and describes some basic procedures for preprocessing a collection of documents. Definition and representation of network entities is presented in the fourth section. The fifth section explains what types of distance measures can be used with network entities or documents. The next section suggests some tips and practices to be followed when deciding which relations are appropriate for the generated BisoNet. Use case about autism is presented in the seventh section. The last section sketches our plans for future work in the Bison project. Acknowledgements and references are listed at the end of this paper.

2 Problem Description: Creation of BisoNets from Text

When creating large bisociation networks (BisoNets) from texts, we have to address the same two issues as in network creation from any other source: define a method for identifying entities, and define a method for discovering relations between these entities. Since text documents can be acquired from very diverse sources we can apply very diverse techniques to generate BisoNets.

In practice, a workflow for converting a set of documents into a BisoNet is more complex than just identifying entities and relations. We have to be able to preprocess text and filter out noise, to generate a large number of in-memory entities and calculate various distance measures between them effectively. As these tasks are not just conceptually difficult, but also computationally very intensive, a great care is needed when designing and implementing algorithms for BisoNet construction.

[2] PubMed database: http://www.ncbi.nlm.nih.gov/pubmed.
[3] Medical Subject Headings: http://www.ncbi.nlm.nih.gov/sites/entrez?db=mesh.

The currently proposed “text to BisoNet” system, called Texas (Text Assistant), consists of the following modules:

- connect to a data source and collect a set of documents,
- preprocess the documents,
- define network entities (considering background knowledge),
- search / count entities in the text and create the in-memory entity representation,
- define and calculate various measures of similarities/distances between entities,
- establish relations between entities using the calculated measures, and
- output the created BisoNet.

A sample workflow, as implemented in the Orange4WS extension [Pod09] of the Orange data mining toolbox [Dem04], is illustrated in Figure 1 (BOW=”bag of words” representation of documents).

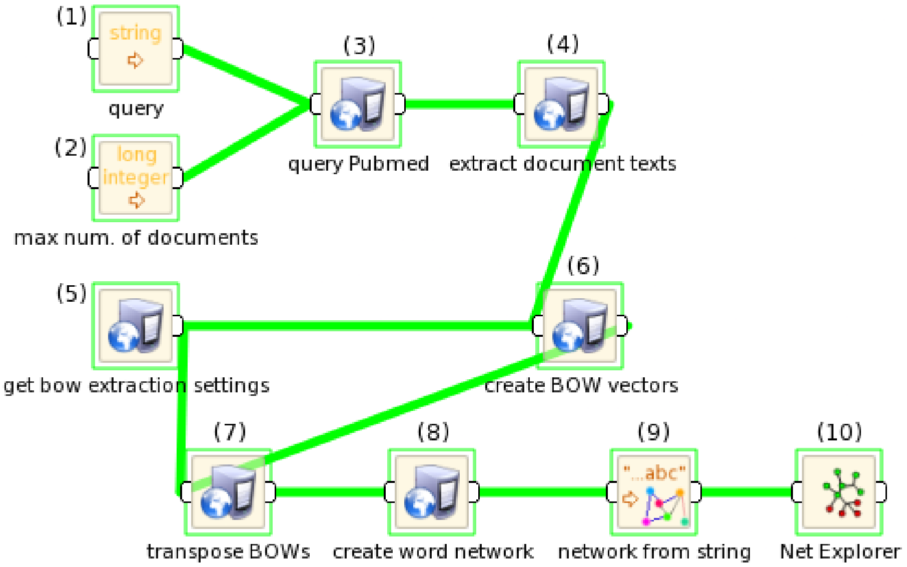

Figure 1: A workflow of text mining algorithms and services.

This paper describes the specific issues that arise when dealing with texts and which can usually not be applied directly to other kinds of databases. The described Texas implementation is built on top of the LATINO[4] library of link analysis and text mining software. This library contains a majority of elementary text mining procedures, but, as the creation of BisoNet is a very specific task (in the text mining world), a lot of modules had to be implemented from scratch or at least optimized considerably.

[4] LATINO library: http://sourceforge.net/projects/latino/.

3 Acquiring a Text Corpus and Creating a BisoNet

This section briefly describes the first and the last step in the workflow of BisoNet creation, i.e., connecting to a data source to collect the documents and the output of the created BisoNet. Since these two issues are mainly technical - they are neither difficult nor computationally expensive - we here only list what our implementation supports and what are the options to be considered.

As there are no standards about the text interchange format in the BISON project and for the sake of simplicity we currently accept just textual and XML files as an input to the procedure. In the future, we can simply add also the following alternatives:

- acquiring documents using soap web services (e.g.: PubMed uses soap web service interface to access their database),
- selecting documents from various SQL bases,
- crawling the internet and gathering documents from web pages. (e.g.: Wikipedia articles), and
- collecting documents from snippets returned from search engines (e.g.: Google snippets).

We have provided the output of the created BisoNets in two different formats:

- the Biomine[5] network file format, used in the Biomine Knowledge discovery in biological databases project [Sev06],
- the Pajek[6] network file format, used in the Pajek program for large network analysis [Bat03],

enabling BisoNet visualization and analysis with Biomine and Pajek, respectively.

In addition to explaining various aspects of preprocessing, this section also briefly describes basic text mining concepts and terminology, some of which are taken from [Fel07].

Preprocessing is the most important part of network extraction from text documents. Its main task is the transformation of unstructured data from text documents into a predefined well-structured document data representation. As shown below, preprocessing is inevitability very tightly connected to the extraction of network entities. In our case, actual network entities are totally defined after preprocessing is finished. The only thing we can later do is to remove some of the useless entities from the set.

In general, the task of preprocessing consists of the extraction of documents' features from documents. The set of all features from document collection is called a representational model. Each document can be presented as a subset of features that it contains. If we write these features of every document in the form of a vector we get the most standard document representation called feature vectors. Given that one of

[5] Biomine project: http://www.cs.helsinki.fi/group/biomine/.

[6] Pajek program: http://pajek.imfm.si/doku.php.

the characteristics of documents' feature vectors is their sparseness, they are often referred also as sparse vectors. In short, the goal of preprocessing is to extract a sparse feature vector for each document from the given document collection.

Commonly used document features are characters, words, terms and concepts [Fel07]. Characters and words carry little semantic information and are therefore not interesting to consider. On the other hand, terms and concepts carry much more semantic information. Terms are usually considered as single or multiword phrases selected from the corpus by means of term-extraction mechanisms (e.g. because of their high frequency) or are present in an external lexicon of a controlled vocabulary. Concepts or keywords are features generated for documents employing the categorization or annotation of documents. Common concepts are derived from manually annotating a document with some predefined keywords or by inserting a document into some predefined hierarchy. When we refer to document features, we mean terms and concepts that we were able to extract from the documents.

Since high-quality features are hard to acquire, all possible methods that could improve this process should be used at this point. The general approach that usually helps the most is achieved by incorporating background knowledge about the documents and their domain. The most elegant technique to incorporate background knowledge is the use of a controlled vocabulary. Controlled vocabulary is a lexicon of all relevant terms that exist in a given domain. Here we can see a major difference when processing general documents as compared to scientific documents. For many scientific domains there exists not only a controlled vocabulary but also a lot of documents inside scientific article collections are pre-annotated. In this case we can quite easily create feature vectors since we have terms as well as concepts already pre-defined. We just have to find them in the documents. Other interesting approaches to identifying concepts include methods such as KeyGraph [Ohs98], which extract keywords/concepts with minimal assumptions or background knowledge, even from individual documents.

A standard collection of preprocessing techniques [Fel07] is listed below, together with a set of functionalities implemented in our system contains.

- Tokenization: continuous character stream must be broken up into meaningful sub-tokens, usually words or terms in case where a controlled vocabulary is present. Our system uses a standard unicode tokenizer: it partly follows the Unicode Standard Annex #29[7] for Unicode Text Segmentation. The alternative is a more advanced tokenizer which tokenizes strings according to a predefined controlled vocabulary and discards all the other words/terms. Such a tokenizer was used in the test scenario of BisoNet creation from PubMed documents described in Section 8.
- Stopword removal: stopwords are some predefined words from a language that usually carry no relevant information (e.g.: and, or, a, an, ... in English); the usual practice is to ignore them when building a feature set. Our implementation uses a predefined list of stopwords - some common lists that

[7] Unicode Standard Annex #29: http://www.unicode.org/reports/tr29/#Word_Boundaries.

are already included in the library are taken from Snowball[8] - a small string processing language designed for creating stemming algorithms.

- Stemming or lemmatization: the process that converts each word/token into the morphologically neutral form. The following alternatives have been made available: Snowball stemmers, the Porter stemmer [Por80], Lemmagen lemmatizer [Jur07].
- Part-of-speech (POS) tagging: the annotation of words with the appropriate POS tags based on the context in which they appear.
- Syntactical parsing: performs a full syntactical analysis of sentences according to a certain grammar. Usually shallow (not full) parsing is used since it can be efficiently applied to large text corpora.
- Entity extraction: methods that indentify which terms should be promoted as entities and which not. Entity extraction through words grouping into terms using n-gram extraction mechanisms (an n-gram is a sub-sequence of n items from a given sequence) has been implemented.

4 Network Entities

The design choice of our approach is that the entities of the BisoNets will be directly the features of documents, i.e., the terms and concepts, described in the previous section. The following steps are independent of how terms and concepts have actually been identified.

After entities definition one also has to provide some representation of entities in a way which enables efficient calculation of distance measures between them. In the same way as documents are represented as sparse vectors of features (entities), also entities can be represented as sparse vectors of documents. This is illustrated in Example 1: if entity ent_1 is present in documents doc_1, doc_3 and doc_4 then its feature vector would consist of all these documents (with appropriate weights). By analogy to the original vector space - feature space, the newly created vector space is called the document space. While documents “live” in the feature (entity) space, the entities “live” in the document space.

Note that if we write document vectors in the form of a matrix, than the conversion between the feature space and the document space is performed by just transposing the matrix (see Example 1). The only question that remains open for now is what to do with the weights? Is weight $w^f_{x:y}$ identical to weight $w^d_{y:x}$? This depends on various aspects, but mostly on how we define weights of the entities (features) in the first place (when defining document vectors.)

There are four most common weighting models for assigning weights to features:

- Binary: feature weight is either one, if the corresponding feature is present in the document, or zero otherwise.

[8] Snowball: http://snowball.tartarus.org.

- Term occurrence: feature weight is equal to the number of occurrences of this feature.
- Term frequency: weight is derived from the term occurrence by dividing the vector by the sum of all the weights (number of all the features) – it can be also viewed as term occurrence normalized by the Manhattan length of the vector.
- TF-IDF: Term Frequency-Inverse Document Frequency is the most common scheme for weighting features. It is defined as: $w_{x:y}^{TFIDF} = \mathrm{TermFreq}(ent_x, doc_y)\log\left(\frac{N}{DocFreq\ (ent_x)}\right)$, where $\mathrm{TermFreq}(ent_x, doc_y)$ is the frequency of feature ent_x inside document doc_y, N is the number of all documents and $DocFreq(ent_x)$ is the number of documents that contain ent_x. The idea behind TF-IDF measure is to lower the weight of features that appear in many documents.

Documents	Extracted entities
doc_1	ent_1, ent_2, ent_3
doc_2	ent_3, ent_4, ent_4
doc_3	ent_1, ent_2, ent_2, ent_5
doc_4	ent_1, ent_1, ent_1, ent_3, ent_4, ent_4

Original documents and extracted entities

Feature space	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$w^{f}_{1:1}$	$w^{f}_{1:2}$	$w^{f}_{1:3}$		
doc_2			$w^{f}_{2:3}$	$w^{f}_{2:4}$	
doc_3	$w^{f}_{3:1}$	$w^{f}_{3:2}$			$w^{f}_{3:5}$
doc_4	$w^{f}_{4:1}$		$w^{f}_{4:3}$	$w^{f}_{4:4}$	

Sparse matrix of documents: $w^{f}_{x:y}$ denotes the weight (in the feature space) of entity y in the feature vector of document x

Document space	doc_1	doc_2	doc_3	doc_4
ent_1	$w^{d}_{1:1}$		$w^{d}_{1:3}$	$w^{d}_{1:4}$
ent_2	$w^{d}_{2:1}$		$w^{d}_{2:3}$	
ent_3	$w^{d}_{3:1}$	$w^{d}_{3:2}$		$w^{d}_{3:4}$
ent_4		$w^{d}_{4:2}$		$w^{d}_{4:4}$
ent_5			$w^{d}_{5:3}$	

Sparse matrix of entities: $w^{d}_{x:y}$ denotes the weight (in the document space) of document y in the document vector of entity x

Example 1: Conversion between the feature and the document space.

These four methods can be further modified with vector normalization (dividing each vector so that length - usually the Euclidian or Manhattan length - of the vector is 1). If and when this should be done depends on several reasons: one of them is also the decision which distance measure one will use in the next step – the relation identification step. If cosine similarity is used, it actually does not matter if the vectors are pre-normalized, as this is also done during distance calculation. Example 2 shows the four measures in practice – documents are taken from Example 1. Weights are calculated for the feature space and are not normalized.

For testing purposes we have implemented all four weighting models so one can experiment which is the most suitable to some domain. It is also up to workflow designer to decide whether vectors should be normalized or not. Currently we are still researching what to do with weights when we are transforming back and forth between feature space and document space. At this point we leave this decision also to a workflow designer and support three most sensible approaches:

- Leave weights unchanged.
- Leave weights unchanged but normalize the entities vectors after transformation.
- Recalculate all weights according to the new space.

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	1	1	1		
doc_2			1	1	
doc_3	1	1			1
doc_4	1		1	1	

Binary weight

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	1	1	1		
doc_2			1	2	
doc_3	1	2			1
doc_4	3		1	2	

Term occurrence

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$^1/_3$	$^1/_3$	$^1/_3$		
doc_2			$^1/_3$	$^2/_3$	
doc_3	$^1/_4$	$^2/_4$			$^1/_4$
doc_4	$^3/_6$		$^1/_6$	$^2/_6$	

Term frequency

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$(^1/_3)\cdot\log(^4/_3)$	$(^1/_3)\cdot\log(^4/_2)$	$(^1/_3)\cdot\log(^4/_3)$		
doc_2			$(^1/_3)\cdot\log(^4/_3)$	$(^2/_3)\cdot\log(^4/_2)$	
doc_3	$(^1/_4)\cdot\log(^4/_3)$	$(^2/_4)\cdot\log(^4/_2)$			$(^1/_4)\cdot\log(^4/_1)$
doc_4	$(^3/_6)\cdot\log(^4/_3)$		$(^1/_6)\cdot\log(^4/_3)$	$(^2/_6)\cdot\log(^4/_2)$	

TF-IDF: term frequency – inversed document frequency

Example 2: Weighting models of features in document vectors (from Example 1).

It is worthwhile to notice again the analogy between the feature space and the document space. Although we have developed the methodology for entities network extraction, the developed approach can be used also for document network extraction.

Moreover, both approaches can be used to extract the same network where documents and entities are connected using some special relations.

5 Distance Measures between Vectors

This section describes some distance measures between vectors in either the feature space or the document space. The choice of a preferable distance measure should be tightly connected to the choice of the weighting model. Some of the combinations are very suitable for each other and may even have some understandable interpretation or experimentally evaluated important value, while others may be less appropriate combination pairs. Therefore we also list commonly used pairs of weighting model and distance measure and describe them.

Our implementation is optimized to the calculation of lengths of sparse vectors: $|vec_x|$ and dot products between those vectors: $\mathrm{DotProd}(vec_x,\ vec_y)$. For that reason, we state also how different distance measures are expressed using these two calculations (if applicable for the described measure).

The most common measures in vector spaces, which are also implemented in our system, are the following:

- Dot products: $\mathrm{DotProd}(vec_x,\ vec_y)$.
- Cosine similarity: which is actually dot product normalized by the length of both vectors $\mathrm{CosSim}(vec_x,\ vec_y) = \frac{\mathrm{DotProd}\ (vec_x,\ vec_y)}{|vec_x||vec_y|}$. In the cases where vectors are already normalized, cosine similarity is identical to the dot product.
- Jaccard index: this similarity coefficient measures the similarity between sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets:
  $\mathrm{JaccInx}(vec_x,\ vec_y) = \frac{|vec_x \cup vec_y| - |vec_x \cap vec_y|}{|vec_x \cup vec_y|} = \frac{\mathrm{DotProd}\ (vec_x,\ vec_y)}{|vec_x|+|vec_y|-\mathrm{DotProd}\ (vec_x,\ vec_y)}$,
  where lengths $|vec_x|$ and $|vec_y|$ are manhattan lengths of these vectors.
- Bisociation index: is the similarity measure defined for the needs of the BISON project. It is explained in more detail in [Bor09]. This measure cannot be expressed by dot product, therefore, the following definition uses the notation derived from Example 1:
  $\mathrm{BisInx}(vec_x,\ vec_y) = \sum_{i=0}^{M} \left(\sqrt[k]{w_{x:i} w_{y:i}} \left(1 - \frac{|\tan^{-1}(w_{x:i}) - \tan^{-1}(w_{y:i})|}{\tan^{-1}(1)}\right)\right)$,
  where *M* is the number of all entities.

Pairs of weighting models for features/entities and distance measures that are usually used together in vector spaces are the following:

- TF-IDF weighting, cosine similarity – this is probably the most commonly used combination for computing similarity in the feature space.
- Binary weighting, dot product – if used in the document space the result is the co-occurrence measure which counts the number of documents where two entities appear together. This is probably the most widely used measure in the document space.
- Term occurrence weighting, dot products – this is another measure of concurrence of entities in same documents. Compared to the previous measure, this one considers also multiple co-occurrence of two entities inside a document and gives them a greater weight in comparison with the case were each appears only once inside the same document.
- Binary weighting, Jaccard index – Jaccard index is defined on the domain of sets, therefore the only reasonable weighting model to use with it is the binary weighting model (since every vector then represents a set of features).
- Term frequency, "Bisociation index" – since Bisociation index was designed with the term frequency weighting in mind, it seems reasonable, to firstly try this combination when determining the weighting model for the Bisociation index.

6 Relations between Entities

At this point a workflow designer has all the required ingredients to create a BisoNet: definition of the entities and the means to calculate distances/similarities between them. This section describes some design techniques to be considered when deciding which of the many possible relations should be included in the network.

Ideas for some of the described approaches were drawn from [Swa06] and its descendant [Pet09]. The main idea of these two articles is to exploit weak relations between entities. This is an innovative and promising attempt to finding interesting – hidden – relations between entities. Hence, we try to simulate this procedure and recreate interesting discoveries made with those algorithms. Consequently, we were encouraged to include also information of weak links into our BisoNet creation procedure.

A common and generally good practice to be followed when creating relations is to annotate them with different types if they are derived using different approaches. In the case one follows this idea, the algorithms of the next step (searching through BisoNets) will have much easier tasks to solve. In such a way one also does not need to worry so much if some relations are unnecessarily defined twice (if the same information comes up using two different techniques), since relations are not merged together but are distinguished by the following algorithms.
We have implemented the following relations/links identifying techniques:

- Strongest links extraction: go through all combinations of pairs of nodes and find the strongest links (usually this means to find relations between most similar entities.) We see at least three options how to accomplish this:
    - The first option is to extract the n strongest links in the whole network.
    - The second option is to extract the m strongest links for every node in the network.
    - The third option is the combination of the first and the second. Retrieve the n strongest links in general and append the m strongest links for each node (if they do not already exist). In this way, the network is connected – every node has minimally m connections, but “stronger” nodes get the opportunity to get better connected than the others.
- Weakest links extraction: find links that have weight more than zero (they exist) but are the weakest among all the links.
    - The three options described in the strongest links extraction can be also applied here.
- Adding links from background knowledge. In the case where we have some background knowledge that already contains links between entities (e.g.: MeSH thesaurus in the case of PubMed articles) we should consider adding them also to the output network.
- Adding inverse vectors. If we are building a network of entities there is also the possibility of adding documents as nodes in the network. Links between entities can be added using numerous described ways, while the relation between entities and documents could be of type “document contains entity”. The same conclusion is valid if we are creating a document network – we can add entities. One concern here can be the great number of links added with this approach; however, some filtering techniques may be applied.

Which of these techniques are appropriate and which are not can only be evaluated using advanced BisoNet search/crawler/exploration algorithms and tools. Given that there are many possible combinations of relations to include in the network, also promising subsets should be identified. So far we did not research this issue, as it is conceptually a separate process – compared to generating BisoNets from text documents. In view of the fact that only results from these algorithms will be able to evaluate the entire process of network creation, this is one of the most important items on our future work agenda.

7 The Autism Case Study

The goal of this use case was to construct a BisoNet from PubMed articles on autism. Autistic disorder (also called autism; more recently described as "mindblindedness") is a neurological and developmental disorder that usually appears during the first three years of life. A child with autism appears to live in his/her own world, showing little interest in others, and a lack of social awareness. Autistic children often have problems in communication, avoid eye contact, and show limited attachment to others. However, many persons with autism excel consistently on certain mental tasks (i.e., counting, measuring, art, music, memory).

We applied the above described Texas process to obtain a BisoNet for autism. We retrieved articles about autism from the PubMed database, identified entities in them using the MeSH vocabulary, and derived co-occurrence relations between entities. A part of the resulting BisoNet, as visualized by the Biomine visualization engine [Sev06], is shown in Figure 2.

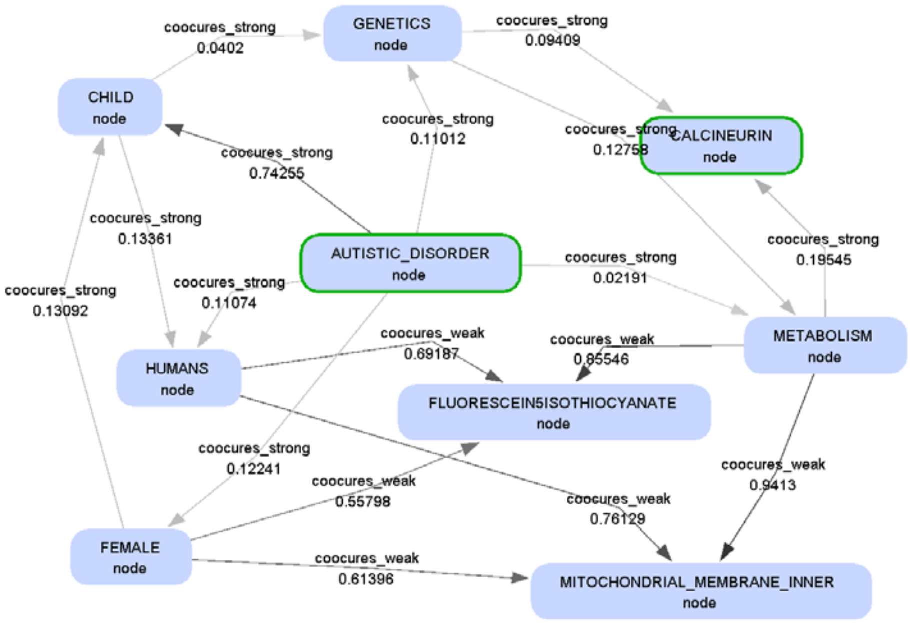

Figure 2: Part of BisoNet, created from PubMed articles on autism.

The cause of autism is not known. Research suggests that autism is a genetic condition, as evidenced by a link between autism and genetics in the BisoNet of Figure 2. It is believed that several genes are involved in the development of autism. Research studies in autism have found a variety of abnormalities in the brain structure and chemicals in the brain; however, there have been no consistent findings. The BisoNet of Figure 2 suggests possible relationships to calcineurin and fluorescensisohticyanate. Ideally, through BisoNet exploration, we hope to discover some still unknown links in this domain.

A part of the BisoNet, created from the PubMed articles on autism, as visualized by the Biomine visualization engine [Sev06], is shown in Figure 2.

8 Future Work

The methodology for creating BisoNets from text, presented in this paper, will be used as a foundation for our forthcoming research on case studies investigated in the BISON project, which include the use of texts in BisoNets. These case studies (benchmarks) will help us not only to validate this methodology, but also to get the overall view of the progress we are doing on bisociation discovery (the core of the BISON project).

The case studies we plan to address using the developed methodology are:

- Migraine treatment and unknown facts detection from the selection of documents out of the PubMed database. The goal of this benchmark is to recreate the Swanson's approach [Swa06] to literature-based discovery of hidden relations between concepts A and C via intermediate B-terms. If there is no known direct relation A-C, but there are published relations A-B and B-C one can hypothesize that there is a plausible, novel, yet unpublished indirect relation A-C. The result of [Swa06] that we want to rediscover is a bisociative link between migraine and magnesium, which was previously unknown.
- Discovery of interesting (previously unstudied) specifics in the domain of autism from the selection of documents out of the PubMed database. This benchmark is about reconstructing the RaJoLink approach [Pet09] to literature-based open discovery process. The Swanson's approach implements closed discovery, the A-B-C process, where A and C are given and one searches for intermediate B concepts. In open discovery, in contrast, only A is given. The RaJoLink idea is to find C via B terms which are rare (and therefore potentially interesting) in conjunction with A.
- Cross contexts (domain) bisociation link discovery in the 20 newsgroups data set[9]. In this setting we want initially to find some mappings between the entities from one domain and equivalent entities from another domain. After identification of such connections, we will try to find bisociations between whole concepts among domains. These bisociations can indicate how to apply solutions of problems from one domain to the open problems of another domain.

We expect that the most time-consuming task during the creation of BisoNets for the above presented case studies will be the definition of the numerous setting at each step of the network creation workflow. Although this paper leaves many such topics unanswered, decisions will have to be made and supported by reasonable arguments.

We will also investigate alternative methods for identifying concepts and discovering relationships between them. In particular, we would like to be able to identify rare but important relationships and separate them from common relationships, even when they are strong. This would give further support to discovery of novel and non-trivial links.

[9] The 20 newsgroups data set: http://people.csail.mit.edu/jrennie/20Newsgroups/.

Acknowledgement

The work presented in this paper was supported by the Slovenian Research Agency grant Knowledge Technologies, and by the grant of the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898. We thank Kimmo Kulovesi for his help and collaboration with using Biomine.

References

Albert, R., Barabasi, A.L.: Statistical mechanics of complex networks. In: Rev Mod Phys, vol. 74(1), pp. 47--97 (2002)

Bales, M.E., Johnson., S.B.: Graph theoretic modeling of large-scale semantic networks. In: Journal of Biomedical Informatics, vol. 39(4), pp. 451--464 (2006)

Batagelj, V., Mrvar, A.: Pajek - Analysis and Visualization of Large Networks. In: Graph Drawing Software, pp. 77--103, (2003)

Berthold, M.R., Dill, F., Kötter, T., Thiel, K.: Supporting Creativity: Towards Associative Discovery of New Insights. In Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD-2008, LNAI 5012, pp. 14--25, (2008)

Borgelt, C., et al.: BISON project Deliverable D2.1: Network Elements. (to appear 2009)

Bostrom, H., et al.: On the definition of information fusion as a field of research. In: Technical report, University of Skovde, School of Humanities and Informatics, Skovde, Sweden (2007)

Demšar, J., Zupan, B., Leban, G.: Orange: From experimental machine learning to interactive data mining. White Paper (2004)

Dura, E., Gawronska, B., Olsson, B., Erlendsson, B.: Towards Information Fusion in Pathway Evaluation: Encoding Relations in Biomedical Texts. In: Proceedings of the 9th International Conference on Information Fusion (2006)

Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2007)

Juršič, M., Mozetič, I., Lavrač., N.: Learning Ripple Down Rules for Efficient Lemmatization. In: Proceedings of the 10th International Multiconference Information Society 2007, vol. A, pp. 206--209 (2007)

Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: Automatic Indexing by Co-occurrence Graph based on Building Construction Metaphor. In: Proceedings of the Advances in Digital Libraries Conference (ADL), pp. 12--18 (1998)

Petrič, I., et. al.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. In: Journal of Biomedical Informatics, vol. 42(2), pp. 219--227 (2009)

Podpečan, V., Žakova, M., Lavrač, N.: Towards a Service-Oriented Knowledge Discovery Platform. In: Proceedings of the Second Service-oriented Knowledge Discovery Workshop at ECML/PKDD - in review (2009)

Porter, M.F.: An algorithm for suffix stripping. In: Program, vol. 14(3), pp. 130--137 (1980)

Racunas, S., Griffin, C.: Logical data fusion for biological hypothesis evaluation. In: Proceedings of the 8th International Conference on Information Fusion (2005)

Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: Proceedings of 3rd International Workshop on Data Integration in the Life Sciences (2006)

Smirnov, A., Pashkin, M., Shilov, N., Levashova, T., Krizhanovsky, A.: Intelligent Support for Distributed Operational Decision Making. In: Proceedings of the 9th International Conference on Information Fusion (2006)

Swanson, D.R., Smalheiser, N.R., Torvik, V.I.: Ranking Indirect Connections in Literature-Based Discovery: The Role of Medical Subject Headings (MeSH). In Journal of the American Society for Information Science and Technology, vol. 57, pp. 1427--1439 (2006)