Automatic Thesaurus Construction System “AUTHECO”

Objective

Automation the process of domain-specific thesaurus construction. Making easier the process, reducing cost and duration of the process.

Destination of the System

System is designed for the knowledge management professionals. It’s goal is to make the process of thesauri construction easier and more flexible by means of analyzing of the domain text corpus. The proposed system help to an expert by detecting words, multiword expressions and relations between them to be included into thesaurus.

Thesaurus in an Information Storage and Retrieval System

![Diagram of a system showing the interaction between users, search engine, index, documents, and thesaurus.](image-url)
Manual and semi-automatic technology of thesauri construction

Manual technology

Linguistic resources, standards etc  
Text corpus  
Expert  
Thesaurus

Problems
1. High cost of thesaurus construction
2. Long duration of construction (~1 year)
3. Quality of thesaurus slightly depends on expert qualification
4. Not possible to analyze full corpus manually
5. Low flexibility of thesaurus construction process

Automatic technology

Linguistic resources, standards etc  
AUTHECO  
Text corpus  
Expert  
Thesaurus

Editing results of automatic corpus processing; Providing information about corpus for making decisions during thesaurus construction.

Solutions of the problems
1, 2: Reducing costs and duration of development by easing the process of thesauri construction.
3: Reducing influence of expert qualification on quality of thesaurus by using robust statistical data for constructing a semantic resource.
4: Automatic processing of whole text corpus, providing robust statistics about words and multiword expressions.
5: Possibility to re-run corpus analyzing procedures
## Comparative analysis of analogous systems

**XIP** – Xerox Incremental Parser  
**Oracle Text** – Oracle Text platform with Oracle Thesaurus Management System extension  
**GALILEI** – Text Mining platform (ULB open-source project)  
**a.k.a.** – Synercon Management Consulting a.k.a.®

### Comparative analysis of functions

<table>
<thead>
<tr>
<th>Function</th>
<th>K_{36}</th>
<th>XIP</th>
<th>Oracle Text</th>
<th>GALILEI</th>
<th>a.k.a.</th>
<th>AUTHECO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>0.15</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>—</td>
<td>+</td>
</tr>
<tr>
<td>Extracting MWE</td>
<td>0.15</td>
<td>+</td>
<td>—</td>
<td>+</td>
<td>—</td>
<td>+</td>
</tr>
<tr>
<td>Building set of thesaurus descriptors</td>
<td>0.15</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>+</td>
</tr>
<tr>
<td>Finding synonymy relationships</td>
<td>0.05</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>+</td>
</tr>
<tr>
<td>Finding relationships between thesaurus descriptors</td>
<td>0.2</td>
<td>+</td>
<td>+</td>
<td>—</td>
<td>—</td>
<td>+</td>
</tr>
<tr>
<td>Corpus documents clustering</td>
<td>0.1</td>
<td>—</td>
<td>+</td>
<td>—</td>
<td>—</td>
<td>+</td>
</tr>
<tr>
<td>Means for storing and editing a thesaurus</td>
<td>0.1</td>
<td>—</td>
<td>+</td>
<td>—</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Summ = 0.5, 0.55, 0.4, 0.2, 1

### Comparative analysis of data processing methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Preprocessing</th>
<th>Extracting MWE</th>
<th>Building set of thesaurus descriptors</th>
<th>Finding synonymy relationships</th>
<th>Finding relationships between words and expressions</th>
<th>Documents clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>XIP</td>
<td>Lemmatization, POS-tagging, Syntactic relations</td>
<td>N-gramms, special grammars etc. Linguistic resources</td>
<td>—</td>
<td>—</td>
<td>Distributive analysis, Data Mining, Apriori algorithm</td>
<td>—</td>
</tr>
<tr>
<td>Oracle Text</td>
<td>Lemmatization, correcting misprints</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>Data Mining: Apriori algorithm, Association Rules etc</td>
<td>Enhanced K-Means, Orthogonal Portioning Clustering</td>
</tr>
<tr>
<td>GALILEI</td>
<td>Lemmatization</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>a.k.a.</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>AUTHECO</td>
<td>Lemmatization, POS-tagging, Syntactic relations</td>
<td>Unitex graphs and dictionaries</td>
<td>Selection function based of statistical and linguistic data</td>
<td>Using base of synonyms</td>
<td>Distributive analysis</td>
<td>Latent Semantic Analysis + K-Means</td>
</tr>
</tbody>
</table>
**AUTHECO system architecture**

- **GUI – Windows.Forms**
  - Data
  - Interface events
  - System functional core – .NET assembly
    - Module for constructing set of preferable descriptors
    - Module for synonymy detection
    - Module for distributive analysis
  - Data manipulation
  - Data access components – ADO.NET

- **Syntax analyzer**
  - Cordial Analyzer (Win32 application)
  - GUI commands
  - Text files

- **Windows-platform** (client application)
  - Interaction with the system: GUI commands

- **Stop lists** (Text files)
- **Text corpus** (XML-files)
- **Commands**
- **Text files**
- **SVDPACKC** (console Win32 application)
  - Module for singular vector decomposition (SVD)
- **Lemmatizer and POS-tagger**
- **TreeTagger** (console Win32 application)
- **Multiword expressions detecting module**
- **Unitex 3 programs + graphs and dictionaries** (console Win32 Applications + binary files)
- **Developed unique modules**

- **DBMS MySQL 5**
- **DBMS Microsoft SQL Server 2005**
- **Thesaurus EuroWordNet, Thesaurus STRATEGO**
- **Linguistic database AUTHECO**
- **DMX queries**

- **Clustering module**
  - Analysis Services 2005
  - Microsoft Clustering Algorithm

- **Interaction with the system:**
  - GUI commands
  - Text files

**Modules:**
- **Command**
- **Text files**
- **Stop lists** (Text files)
- **Text corpus** (XML-files)
- **Module for constructing set of preferable descriptors**
- **Module for synonymy detection**
- **Module for distributive analysis**
- **SVDPACKC** (console Win32 application)
  - Module for singular vector decomposition (SVD)
- **Lemmatizer and POS-tagger**
- **TreeTagger** (console Win32 application)
- **Multiword expressions detecting module**
- **Unitex 3 programs + graphs and dictionaries** (console Win32 Applications + binary files)
- **Developed unique modules**

**Integration points:**
- **GUI commands**
- **Text files**
- **SQL queries**
- **XML-files**
- **Graphs and dictionaries**
- **Win32 applications**
- **.NET assembly**
Separation of manual and automatic data processing in the technology

Text corpus

Corpus preprocessing

Splitting corpus documents into thematically related subsets (clusters)
Editing clusters (adding/removing documents), assigning labels to clusters

Building set thesaurus descriptors
Editing set of thesaurus descriptors

Building set of synonymy (USE) relationships between descriptors
Editing set of synonymy relationships. Adding domain specific relation.

Building set of associative (RT) and hierarchical relationships between descriptors
Assigning type for all acquired relations. Deleting all relations except RT and NT. Editing set of hierarchical (NT) and associative (RT) relationships,

Automatic processing by the system
Manual processing by an expert

Thesaurus
Technology of semi-automatic thesauri construction

1. Corpus preprocessing
   - Text corpus
   - Stop list
   - Lemmatizer and POS-tagger
   - Multiword expressions extraction module

   - Set of words and expressions, statistical data about the corpus

2. Building set of thesaurus descriptors
   - Set of words and expressions, statistical data about the corpus
   - ISO, ANSI standards

3. Building set of synonymy (USE) relationships between descriptors
   - Set of synonymy relationships between descriptors of thesaurus
   - Base of synonyms

4. Building set of associative (RT) and hierarchical relationships between descriptors
   - Set of NT and RT relationships between descriptors of thesaurus
   - Syntactic analyzer

   - Set of NT and RT relationships between descriptors of thesaurus

   - Expert

   - Editing clusters, assigning labels to clusters
   - Editing set of thesaurus descriptors
   - Editing synonymy relationships
   - Editing set of hierarchical and associative relationships
   - Editing constructed thesaurus

   - Subcorpus 1
   - Subcorpus 2
   - Subcorpus N

   - SVD module
   - Clustering module
Text corpus

Multiword expressions detecting module

Marked text corpus

Tokenizer, Lemmatizer and POS-tagger

Lemmatized tokenized and POS-tagger text corpus

Statistics calculating module

Module for eliminating stop-words

END

In fact, computational linguistics has solved lot of problems...

In fact, \texttt{<MWE>computational linguistics</MWE> has solved lot of problems...}
Building set of thesaurus descriptors (2)

Weighing of words and expressions with selection function:

\[ F_{\text{select}} = \frac{\omega_i}{f_{\text{freq}}(d_i, \text{pos}, \text{freq}, \text{tf} \cdot \text{idf}, \text{mwe}, \text{head})}; \omega_i \in [0; 1] \]

where:
- \( d_i \) — descriptor;
- \( \text{pos} \) — part of speech;
- \( \text{freq} \) — frequency;
- \( \text{tf} \cdot \text{idf} \) — TF*IDF statistical ratio;
- \( \text{mwe} \) — true if descriptor is multiword expression, otherwise -- false;
- \( \text{head} \) — true if descriptor was extracted from header, otherwise -- false.

Function is a combination of constraints on the input variables.
For example, \( f_{\text{freq}} \) has to belong to a predefined interval \( f_{\text{freq}} \in [f_{\text{freq}}_{\text{min}}; f_{\text{freq}}_{\text{max}}] \) in accordance with Zipf law (see figure).
Algorithm of building set of relations between descriptors

Set of thesaurus descriptors – \( D_{pref} = \{d_1, d_2, ..., d_n\} \), where \( n \) – power of set; 
\( d_i \) – a descriptor, represented by vector \( d_i = (w_1, w_2, ..., w_k) \), \( k > 0 \), where \( w_i \) – word or MWE.

\( S_{ewn} \) – set of all synonymy relationships of EuroWordNet thesaurus, \( S_{ewn} = \{s_1, s_2, ..., s_m\} \), where \( s_i \) – synonymy relationship represented with tuple \( (d_i; d_j); d_i \neq d_j \).

Algorithm
Goal of this algorithm is to build subset of set of EuroWordNet synonymy relationships \( S_D = \{s_1, s_2, ..., s_t\}; S_D \subseteq S_{ewn} \).

1. For all \( d_i \) from \( D_{pref} \) find subset \( S_i \subseteq S_{ewn} \) what \( S_i = \{s_{i1}, s_{i2}, ..., s_{it}\} \); \( \forall s_{ij} = (d_i; d_k); j \in [1; t] \), \( d_k \in D_{pref} \).

We get the relationships which contain thesaurus descriptors from \( D_{pref} \).

2. Merge set \( S_i \) with set of synonymy relationship of the thesaurus: \( S_D = S_D \cap S_i \).

3. Go to next descriptor. If all the descriptors were handled then end.
Building set of associative (RT) and hierarchical (NT) relationships between descriptors (4-a)

**4-b**

- **Lemmatized tokenized and POS-tagger text corpus**
- **Syntax analyzer Cordial Analyzer**
- **AUTHECO database**
- **Text corpus**
- **Module for building vector space model of text. Algorithms of distributive analysis first order bag-of-words and syntactic method.**
- **Set of thesaurus descriptors**
- **Set of feature vectors**
- **Module for normalizing set of feature vectors**
- **Set of normalized feature vectors**
- **Module for calculating similarity matrix**
- **Similarity matrix**
- **Module for building set of worth relationships between descriptors**
- **Set of relationships between thesaurus descriptors (without type of relation)**
- **GUI for editing relations between thesaurus descriptors**
- **Final set of relationships between thesaurus descriptors with type of relation**
- **Saving all the data into linguistic database AUTHECO**

**Calculating similarity matrix:**

$$R_{\text{pref}} = \frac{\mathbf{e}_i \mathbf{e}_j}{||\mathbf{e}_i|| \cdot ||\mathbf{e}_j||}$$

where $\mathbf{e}_i$ – feature vector for thesaurus descriptor $d_i$.

*For calculating similarity matrix $S$, we use $\text{cos}$ as a similarity measure between two vectors.*

Thus $S(d_u, d_v) = \text{cos}(\mathbf{d}_u, \mathbf{d}_v) = \frac{\mathbf{d}_u \cdot \mathbf{d}_v}{\|\mathbf{d}_u\| \cdot \|\mathbf{d}_v\|}$

**Matrix normalization**

$$\text{PMI} - \text{Pointwise Mutual Information}$$

$$\text{PMI}(x,y) = \log\frac{P(x,y)}{P(x)P(y)} = \log\frac{\text{freq}(x,y)}{\text{freq}_x \cdot \text{freq}_y}$$

$\text{freq}_i$ – frequency of word of expression $i$.

**For expert**

- **Selecting worth relationships:**
  - We include into the set of relations $R_{\text{pref}}$ relationships $r_i$ with $w_i$ not less than a predefined threshold:
  
  $$r_i \in R_{\text{pref}} : w_i \geq w_{\text{min}}$$

**Editing set of relationships and assigning types of relationships (NT / RT)**
Vector space model in the algorithms of distional analysis (4-b)

First order bag-of-words method

Distributional hypothesis – words with a similar meaning appear in similar contexts

**Syntactic method**

**SRbol** = \{(sr₁, sr₂, ..., srₙ)\} – set of all extracted syntactic relations with syntactic analyzer, and \(srᵢ\) – tuple with format \((dᵢ, typeᵢ, dᵢ')\).

\( SR_{pref} \subseteq SR_{bol}; SR_{pref} = \{(sr \in SR) \land (type \in T_{pref}) \land (dᵢ \in D_{pref})\} \), where \(T_{pref} = \{\text{type}_1, \text{type}_2, \ldots, \text{type}_ₙ\} = \{(\text{subj}, \text{subj-of}, \text{obj}, \text{obj-of}, \ldots, \text{adj}, \text{mod}, \text{mod-of})\} \)

- set of syntactic relations taken into consideration in the model.
- \(T_{pref} \subseteq T\) – множество всех ивключенных типов отношений с использованием синтаксического анализатора.

\( GSR = \{\text{g}_1, \text{g}_2, \ldots, \text{g}_n\} = \{\text{g}_₁ \mid \exists \text{sr} = (dᵢ, typeᵢ, dᵢ') \in SR_{pref}\} \); thus \( GSR \) contain set of elements like \((*, \text{subj},* - \text{of}, \text{work})\).

**Syntactic dependencies in a sentence:**

“**I have a brown dog**”
- (have, subj, I); (I, subj-of, have), (dog, obj-of, have),
- (dog, adj-mod, brown), (brown, adj-mod-of, dog), (dog, det, a),
- (a, det-of, dog) \((d₁, \text{type}_₁, d₁')\).

\(d₁, d₁'\) – are descriptors and \(d₁ \in D_{pref}\)

\(typeᵢ\) – are type of syntactic relation \(typeᵢ \in T\)

End of corpus

\(dᵢ \in D_{pref}\)

**Begin**

\(i = 0\)

\(i = i + 1\)

\(dᵢ \in D_{pref}\)

\(w = i \cdot \text{CONTEXT\_SIZE}\)

\(\text{Add into the feature vector} \ e_w = e_w + 1\)

\(w = w + 1\)

\(i = i + 1\)

\(End\)

**Format of feature matrix** (documents-documents), where \(F_{dᵢ} \) – feature vector of thesaurus descriptor \(dᵢ\)

**Algorithm parameter is context window size (\text{CONTEXT\_SIZE})**.

- 1 word \( \leq \text{CONTEXT\_SIZE} \leq 1 \) paragraph
- \(i\) – Pointer at current descriptor
- \(w\) – Pointer at current word into the context window

Example for context size = 1 word:
- \[I \text{ have a brown dog}\]
- \[I \text{ have a brown dog}\]
- \[I \text{ have a brown dog}\]
- \[I \text{ have a brown dog}\]
- \[I \text{ have a brown dog}\]

\(D_{pref}\)

- \(dᵢ = \text{industry}\)
- \(dᵢ = \text{agriculture}\)
- \(dᵢ = \text{fertilizer}\)
- \(dᵢ = \text{worker}\)

\(D_{pref}\)

- \( \text{industry}\)
- \( \text{agriculture}\)
- \( \text{fertilizer}\)
- \( \text{worker}\)

Begin

\(k = 0\)

\(\text{sr}_k = (d_k, \text{type}_k, d_k')\)

Add to the feature vector \(F_{d_k'}\), in the dimension \(gᵢ = (*, \text{type}_k, d_k')\):

\(\text{Add to the feature vector} \ e_{j_k} = e_{j_k} + 1\)

\(k = k + 1\)

End

All elements of \(SR_{pref}\) were processed

yes

no