Syntactic and semantic aspects of natural language processing

PhD Dissertation

Ph.D Candidate
Anca Dinu

Supervisor
Acad. Prof. Dr. Solomon Marcus
What this theses is about

• The field of this thesis is computational linguistics, natural language processing and formal and computational semantics.
• Hence, it has a profound interdisciplinary character between informatics, linguistics and mathematics.
• It deals with *discourse* phenomena (we use the notion of *discourse* in the sense of multi-sentential text uttered in a natural language).
The structure of the theses

• This work comprises three main parts:
  1. The first part provides an explicit formal account of discourse semantics that extends Barker & Shan’s (2008) sentence level semantics based on computer science notion of ‘continuations’.
  2. The second part presents an on-going research: the construction and annotation of a Romanian Generative Lexicon (RoGL), along the lines of Generative Lexicon Theory of Pustejovsky (2006).
  3. The third part comprises two experiments of classification by coherence/incoherence of short English and Romanian texts, respectively.
Published papers:

Published papers:


Published papers:


The first part provides an explicit formal account of discourse semantics that extends Barker & Shan’s (2008) (sentence level) continuation semantics.

This semantics employs the notion of ’continuations’, categorial grammars and a type shifting mechanism.

It accounts for a wide range of natural language semantic phenomena, such as:

– binding pronominal anaphora (we use the term anaphora in its linguistic sense: an instance of an expression referring to another one, usually located in preceding utterances), quantifier scope, negation, focus, hierarchical discourse structure, ellipsis, accommodation and event quantification.
The discourse semantics we propose is:

- Directly compositional;
- Variable free;
- Dynamic.
- Extensional (but intentionality could be in principle accounted for in this framework);
First part: ‘continuation’ semantics: introduction and motivation

• Historically, the first continuation operators were undelimited (call/cc or J). An **undelimited continuation** of an expression represents “the entire (default) future for the computation” of that expression.

• Felleisen (1996) introduced delimited (or composable) continuations such as control (‘C’) and prompt (‘%’). A **delimited continuation** represents the future of the computation of the expression up to a certain boundary.

• Remarkably, the natural-language phenomena discussed here make use only of delimited continuations.
First part: ‘continuation’ semantics: introduction and motivation

• The basic idea of continuizing a grammar is to provide subexpressions with direct access to their own continuations (future context).

• In order to do this, the denotations of the subexpressions must be modified to take a continuation as an argument.

• A continuized grammar is said to be written in continuation passing style which is in fact a restricted (typed) form of λ-calculus.
If we restrict the context to the sentence limits, the continuation of a lexical entry (word) is the interpretation (denotation or meaning) of the sentence with a slot in the place of the lexical entry (a functor).

For instance, the sentence *John saw Mary*, that has the denotation *saw m j* we have:

<table>
<thead>
<tr>
<th>Denotation</th>
<th>j</th>
<th>m</th>
<th>saw</th>
<th>saw m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Its continuation</td>
<td>(\lambda x.\text{saw } m \ x)</td>
<td>(\lambda y.\text{saw } y \ j)</td>
<td>(\lambda R.\text{R } m \ j)</td>
<td>(\lambda P.\text{P } j)</td>
</tr>
</tbody>
</table>
First part: ‘continuation’ semantics:
introduction and motivation

• This simple example illustrates two important aspects of continuations:

(1) every meaningful subexpression has a continuation;
(2) the continuation of an expression is always relative to some larger expression containing it.
First part: ‘continuation’ semantics:

introduction and motivation

• Thus, when *John* occurs in the different sentences, its continuation changes:

<table>
<thead>
<tr>
<th>sentence</th>
<th>John left yesterday</th>
<th>Mary thought John left</th>
<th>Mary or John left</th>
</tr>
</thead>
<tbody>
<tr>
<td>j’s continuation</td>
<td>( \lambda x.\text{yesterday} \ (\text{left } x) )</td>
<td>( \lambda x.\text{thought} \ (\text{left } x)m )</td>
<td>( \lambda x.\left(\text{left } m\right)\text{or}(\text{left } x) )</td>
</tr>
</tbody>
</table>
First part: ‘continuation’ semantics: introduction and motivation

• One of the main challenges of interpreting a discourse (giving it a compositional semantics) is interpreting cross-sentential anaphora.

• Assigning a first order logical representation to a discourse like “A man came. He whistled.” is problematic.
First part: ‘continuation’ semantics: introduction and motivation

• In \([A \text{ man came}]\) = \(\exists x. (\text{man}(x) \land \text{came}(x))\)
  \(x\) is bound by \(\exists\).

• In \([\text{He whistled}]\) = \(\text{whistled}(x)\).
  \(x\) is free.

• How can we get from these:
  \([A \text{ man came. He whistled.}]\) = \(\exists x. \text{man}(x) \land \text{came}(x) \land \text{whistled}(x)\). ?
First part: ‘continuation’ semantics: introduction and motivation

Various dynamic semantic theories that handle this problem were proposed, such as:

– Kamp & Reyle’s Discourse Representation Theory
– Heim’s File Change Semantics,
– Groenendijk & Stokhof’s Dynamic Montague Grammar and Dynamic Predicate Logic,
– Jacobson’s variable free semantics.
First part: ‘continuation’ semantics: introduction and motivation

- Continuation semantics generalizes some results of traditional semantic theories, for example:
  - The generalized quantifier type from Montague grammar \( \langle\langle e, t \rangle, t, t \rangle \) is exactly the type of quantificational determiners in continuation-based semantics;
  - The \( \langle\langle t, t \rangle, t \rangle \) type of sentences in dynamic semantics is exactly the type of sentences in continuation-based semantics.

- This is by no means a coincidence: MG continuizes only the noun phrase meanings; dynamic semantics continuizes only the sentence meanings, rather than continuizing uniformly throughout the grammar as it is done in continuation-based semantics.
First part: ‘continuation’ semantics: introduction and motivation

We use the tower notation for a given expression, which consists of three levels:

- the top level specifies the syntactic category of the expression coached in categorial grammar;
- the middle level is the expression itself;
- the bottom level is the semantic value.

**syntactic category**

**expression**

**semantic value**
First part: ‘continuation’ semantics: introduction and motivation

• The syntactic categories are written

\[
\frac{A|B}{C}
\]

where A, B and C can be any categories.

• We read this counter clockwise as “the expression functions as a category C in local context and takes scope at an expression of category B to form an expression of category A.”
First part: ‘continuation’ semantics: introduction and motivation

• The semantic value
  \[ \lambda k. f[k(x)] \]

  is written vertically, omitting \( k \), as:
  \[ f[\_\_] \]

  \[ x \]

• Here \( x \) can be any expression, and \( f[\_\_] \) can be any expression with a hole \([ \_\_]\).
• Free variables in \( x \) can be bound by binders in \( f[\_\_] \).
• \( k \) plays the role of future computation or context - the expression’s continuation.
First part: ‘continuation’ semantics: introduction and motivation

• When there is no quantification or anaphora involved, a simple sentence like John left is derived as follows:

\[
\begin{align*}
\text{DP} & \quad \text{DP}\backslash S \quad \quad \quad S \\
\text{John} \quad \text{left} & \quad = \quad \text{John left} \\
\text{j} \quad \text{left} & \quad = \quad \text{left j}
\end{align*}
\]

• As usual, the category under the slash (here DP) cancels with the category of the argument expression, and the semantics is function application.
First part: ‘continuation’ semantics: introduction and motivation

- Quantificational expressions have extra layers on top of their syntactic category and on top of their semantic value, making essential use of the powerful mechanism of continuations.

- For example, the derivation for *Everyone left.* is:

\[
\begin{align*}
S|S & \quad S|S \\
\text{DP} & \quad \text{DP}\backslash S \\
\text{everyone} & \quad \text{left} \\
\forall y. [ ] & \quad [ ] \\
y & \quad \text{left} \\
\frac{S|S}{S} & \quad \text{everyone left} \\
\forall y. [ ] & \quad \text{left } y
\end{align*}
\]
First part: ‘continuation’ semantics: introduction and motivation

Here are the two possible modes of combination (Barker & Shan 2008):

\[
\begin{align*}
  \frac{C|D}{A/B} & \quad \frac{D|E}{B} & \quad \frac{C|E}{A} \\
  \text{left} - \exp & \quad \text{right} - \exp = & \quad \text{left} - \exp \text{ right} - \exp \\
  g[\ ] & \quad h[\ ] & \quad g[h[\ ]] \\
  f & \quad x & \quad f(x)
\end{align*}
\]

\[
\begin{align*}
  \frac{C|D}{B} & \quad \frac{D|E}{B\backslash A} & \quad \frac{C|E}{A} \\
  \text{left} - \exp & \quad \text{right} - \exp = & \quad \text{left} - \exp \text{ right} - \exp \\
  g[\ ] & \quad h[\ ] & \quad g[h[\ ]] \\
  x & \quad f & \quad f(x)
\end{align*}
\]
First part: ‘continuation’ semantics: introduction and motivation

• Comparing the analysis above of *John left.* with that of *Everyone left.* reveals that *left* has been given two distinct values. The first, simpler value is the basic lexical entry; The more complex value is derived through the standard type-shifter Lift, proposed by Partee & Rooth (1983), Jacobson (1999), Steedman (2000), and many others:

\[
\frac{A}{expression} \xrightarrow{Lift} \frac{B|B}{expression} \\
\frac{x}{A} \Rightarrow \frac{[]} {x}
\]

• Syntactically, Lift adds a layer with arbitrary (but matching!) syntactic categories. Semantically, it adds a layer with empty brackets.
First part: ‘continuation’ semantics: introduction and motivation

- To derive the syntactic category and a semantic value with no horizontal line, Barker & Shan (2008) introduce the type-shifter Lower:

\[
\frac{A|S}{S} \\
\text{expression} \xrightarrow{\text{Lower}} \text{expression} \\
\frac{f[ ]}{x} \xrightarrow{A} \frac{f[x]}{}
\]

- Syntactically, Lower cancels an S above the line to the right with an S below the line.

- Semantically, Lower collapses a two-level meaning into a single level by plugging the value x below the line into the hole [ ] in the expression f[ ] above the line.
First part: ‘continuation’ semantics: introduction and motivation

• The third and the last general rule is the Bind rule:

\[
\frac{A|B}{DP} \quad \frac{A|DP \triangleright B}{DP}
\]

expression \( \xrightarrow{\text{Bind}} \) expression

\[
\frac{f[ ]}{x} \quad \frac{f([]x)}{x}
\]

• The DP offers to bind a subsequent anaphoric pronoun by transmitting to it the value \( x \).

• The pronouns have lexical entries of the following form:

\[
\frac{DP \triangleright S|S}{DP} \quad \frac{he}{DP} \quad \frac{\lambda y. [ ]}{y}
\]
First part: ‘continuation’ semantics: introduction and motivation

• We shift from sentence level to discourse level by introducing the semantics of dot “.” as a function that takes two sentence denotations and returns a sentence denotation formed from the conjunction of the two.

\[ S\backslash(S/S) \]
\[ . \]
\[ \lambda r \lambda q. r \land q \]
First part: ‘continuation’ semantics: introduction and motivation

• For two affirmative sentences with no anaphoric relations and no quantifiers, the derivation trivially proceeds as follows:

\[
S \\
\text{John came} \\
\text{came } j \\
S \backslash (S/S) \\
\lambda p \lambda q. p \land q \\
S \\
\text{Mary left} \\
\text{left } m
\]

\[
S = \text{John came} . \text{Mary left} \\
\text{came } j \land \text{left } m
\]
First part: ‘continuation’ semantics: introduction and motivation

- Type shifting and the powerful mechanism of continuations are needed when dealing with long distance dependences as quantifier scope or anaphora binding. For instance, the derivation of *A man came. He whistled.* is:

\[
\begin{align*}
\lambda P. \frac{\exists x. P(x) \land [\,] \quad \lambda P. \frac{\exists x. \text{men}(x) \land [\,] \quad \exists x. \text{men}(x) \land (\, \, x)}{\text{men} \quad \exists x. \text{men}(x) \land [\,] \quad \exists x. \text{men}(x) \land (\, \, x)}{\text{men}}
\end{align*}
\]

\[
\begin{align*}
S \frac{a \text{ man came} \quad \exists x. \text{man}(x) \land (\, \, x) \quad \exists x. \text{man}(x) \land (\, \, x)}{\text{came} \quad \exists x. \text{man}(x) \land (\, \, x) \quad \exists x. \text{man}(x) \land (\, \, x)}{\text{came}}
\end{align*}
\]

\[
\begin{align*}
S \frac{S | DP \Rightarrow S}{} \quad S \frac{S | S \Rightarrow S}{} \quad S \frac{S \Rightarrow S | S}{}
\end{align*}
\]

\[
\begin{align*}
S \Rightarrow S | S \Rightarrow S \quad S \Rightarrow S | S \Rightarrow S \quad S \Rightarrow S | S \Rightarrow S
\end{align*}
\]

\[
\begin{align*}
S \frac{S \Rightarrow S | S \Rightarrow S}{} \quad S \frac{S \Rightarrow S | S \Rightarrow S}{} \quad S \frac{S \Rightarrow S | S \Rightarrow S}{}
\end{align*}
\]

\[
\begin{align*}
S \frac{S \Rightarrow S | S \Rightarrow S}{} \quad S \frac{S \Rightarrow S | S \Rightarrow S}{} \quad S \frac{S \Rightarrow S | S \Rightarrow S}{}
\end{align*}
\]
First part: ‘continuation’ semantics: introduction and motivation

\[
\frac{S \mid S}{S} \\
= \text{a man came. he whistled} \quad \xrightarrow{\text{Lower}} \\
\exists x. \text{man}(x) \land (\lambda y. [\_]x) \\
\text{came } x \land \text{whistled } y \\
\]

\[
\frac{S}{S} \\
= \text{a man came. he whistled} \\
\exists x. \text{man}(x) \land (\lambda y. [\text{came } x \land \text{whistled } y ]x) \\
\]

\[
\frac{S}{S} \\
= \text{a man came. he whistled} \\
\exists x. \text{man}(x) \land (\text{came } x \land \text{whistled } x) \\
\]
First part: ‘continuation’ semantics: negation

• In this framework, we give the following denotation for negation:

\[
\begin{align*}
S|S \\
(DP\backslash S)/(DP\backslash S) \\
\text{not} \\
\neg[]
\end{align*}
\]

• Roughly speaking, we used type shifting to model the impossibility for lexical entries placed inside the scope of negation to bind anaphora outside it: *John does not own a car.*

*It is red.*
Similarly, we proposed the semantics of conditionals:

\[
\frac{S \mid S}{(S \mid S \backslash S) \backslash S}
\]

\[
\frac{\text{if} \quad \lnot [\ ]}{\frac{\frac{\lambda p \lambda q. p \land \lnot [\ ]}{q}}{}}
\]

This denotation is general enough to account for discourse level phenomena in which the scope of the consequent extends over the sentence boundaries, such in:

*If John owns a book on semantics, he uses it. He also recommends it.*
First part: ‘continuation’ semantics: quantifiers

- The lexical entry we proposed for quantificational determiner *no* is:

\[
\begin{align*}
\lambda P. \frac{\neg \exists x. (Px \wedge [\ ])}{x} \\
\text{DP/N} \\
\text{no} \\
\frac{S|S}{DP/N}
\end{align*}
\]

- The impossibility of *No man came. *He whistled. is also straightforwardly accounted for.
First part: ‘continuation’ semantics: quantifiers

• We propose the following lexical entries for plural quantificational determiners (that require plural nouns) *some*, *all* and *most*:

\[
\frac{S|S}{DP} / N^{pl}
\]

\[
\text{some} \\
\lambda P. \quad \exists X. (|X| > 1 \land X = \arg\max_x \{|X:PX \land [ ]\}|)
\]

\[
\text{all} \\
\lambda P. \quad \exists X. (X = \arg\max_x \{|X:PX|\}) \land [ ]
\]
First part: ‘continuation’ semantics

\[
\frac{S|S}{DP} / N^p_l \\
\exists X. (X = \arg\max_x |\{X: PX \land [ ]\}| \land 2|X| \geq |\{x: Px\}|)
\]

\[
\lambda P. \frac{\exists X. (X = \arg\max_x |\{X: PX \land [ ]\}| \land 2|X| \geq |\{x: Px\}|)}{X}
\]

- We discuss distributivity and maximality problems and the complexity of plural referring anaphora (which is unfortunately not parallel with singular anaphora).
First part: ‘continuation’ semantics: hierarchical discourse structure

• In order to handle hierarchical discourse structure we give to the “subordinating dot” “.” (that introduces a new subordinated piece of discourse), the following lexical entry:

\[
S\frac{(S|S)/S}{S}
\]

\[
\lambda p \lambda q. \frac{p \wedge []}{q}
\]

• This accounts for the Right Frontier Constraint (Polanyi 1985): the antecedent of a pronoun in the current sentence must be introduced by the previous utterance or one that dominates it in the discourse structure.
First part: ‘continuation’ semantics: hierarchical discourse structure

- To exemplify the Right Frontier Constraint, we give a discourse example from Asher & Lascarides (2003), with the associated structure:

John had a great evening.
He had a great meal.
He ate salmon\textsubscript{x}.
He devoured cheese.
He then won a dance competition.
\*It\textsubscript{x} was a beautiful pink.
First part: ‘continuation’ semantics: accommodation

• To account for the ‘accommodation’ phenomena, we proposed the following two alternative denotations for the definite article *the*:

\[
\frac{DP^pl \triangleright S|S}{DP/N} \quad \frac{S|S}{DP/N}
\]

\[
\lambda P. \frac{\exists y. P(y) \land []}{y}
\]

\[
\lambda P. \frac{\forall y. P(y) \land []}{y}
\]

• The first one accounts for the regular use, while the second one for the accommodated use (for situations when an anaphoric expression does not find a suitable antecedent in the previous discourse, for instance like in: *The door closed*. which is not usually preceded by *There is a door.* )
First part: ‘continuation’ semantics: focus

- We also introduce the following focus operator:

\[
\frac{S|S}{A} \frac{\lambda x. (\lambda k. k(x) \land \forall y. (y = x \lor \neg (k(y))))(\lambda z. [\ ])}{F/A}
\]

- It accounts for the difference in meaning of examples like:
  a. JOHN saw Mary.
  b. John SAW Mary.
  c. John saw MARY.
- It treats the phenomena of free-focus vs. bound-focus anaphora.
First part: ‘continuation’ semantics: events

• We deal with adverbial quantifiers in event semantics framework (Davidson (1967)) which can be embedded in situations semantics (Barwise (1981))

• For instance, we propose the following denotations of *always* and *never*, that quantify over events or situations:

\[
\frac{S|S}{S_E}
\]

\[
\forall e. \text{RelevEvent}(e) \rightarrow []
\]

\[
\frac{S|S}{S_E}
\]

\[
\neg \exists e. \text{RelevEvent}(e) \land []
\]

• These denotations account for anaphoric reference to events/situations like in:

*John kissed Marry. She liked it. / That was nice of him.*
First part: ‘continuation’ semantics: Restricting the Scope of Clause-Bounded Lexical Entries

• In order to block interpretations in which lexical entries having the scope bounded to their minimal clause (such as not, no, every, each, any, etc.), Barker and Shan (2008) suggest to force their scope closing immediately after the interpretation of their minimal clause, by applying Lower.

• But this strategy leaves the actual mechanism that insures the scope closing unspecified.

• We specify such a mechanism designed to ensure that no lexical entries having the scope bounded to their minimal clause will ever take scope outside.
First part: ‘continuation’ semantics: conclusions

• To conclude:
  – We argued that continuations are a versatile and powerful tool, particularly well suited to manipulate scope and long distance dependencies, phenomena that abound in natural language semantics.
  – No other theory to our knowledge lets indefinites, quantifiers, pronouns and other anaphors interact in a uniform system of scope taking, in which quantification and binding employ the same mechanism.

• We leave for future research:
  – Completing an algorithm that generates all possible interpretations for a given discourse in continuation semantics framework;
  – The possibility to express situation semantics in this framework;
  – The comparison of our approach to anaphora with that from algebraic linguistics.
Second part: Building and annotating Romanian Generative Lexicon

• We present in this part an on-going research: the construction and annotation of a Romanian Generative Lexicon (RoGL), along the lines of Generative Lexicon Theory (GLT) of Pustejovsky (2006).

• In Generative Lexicon Theory semantic types constrain the meaning of other words, for instance the verb *eat* imposes on its direct object the interpretation [[Food]].

• GLT places natural language complexity at lexical level instead of formation rules.
Second part: Building and annotating RoGL

- The theory uses the full predicative decomposition, with an elegant way of transforming the sub-predicates into richer argument typing: Argument Typing as Abstracting from the Predicate:

<table>
<thead>
<tr>
<th>Approach</th>
<th>Type</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atomic</td>
<td>e -&gt; t</td>
<td>( \lambda x. \text{[sleep}(x)\text{]} )</td>
</tr>
<tr>
<td>Predicative</td>
<td>e -&gt; t</td>
<td>( \lambda x. \text{[animate (x) ^ sleep}(x)\text{]} )</td>
</tr>
<tr>
<td>Enriched typing</td>
<td>anim -&gt; t</td>
<td>( \lambda x : \text{anim. [sleep}(x)\text{]} )</td>
</tr>
</tbody>
</table>
Second part: Building and annotating RoGL

• Creating a generative lexicon from scratch for any language is a challenging task, due to complex semantic information structure, multidimensional type ontology, time consuming annotation etc.

• Thus, we took as good examples the existing generative lexicons for other languages such as Italian CLIPS or English BSO (that has an ontology of 144 semantic types).

• RoGL has the following structure:
Second part: Building and annotating RoGL

• The interface and the data base where the annotated lexical entries are stored and processed are hosted on the server of Faculty of Mathematics and Informatics, University of Bucharest: http://ro-gl.fmi.unibuc.ro.

• Some instances of annotation process look like:
Second part: Building and annotating RoGL

<table>
<thead>
<tr>
<th><strong>Romanian GL - Words Input Page</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Word</strong></td>
</tr>
<tr>
<td>![Input field with buttons: Check, New]</td>
</tr>
<tr>
<td>Button: &quot;check&quot; will search the specified word in the database and return the found values.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
</tr>
<tr>
<td>![Input field]</td>
</tr>
<tr>
<td><strong>Definition</strong></td>
</tr>
<tr>
<td>![Input field]</td>
</tr>
<tr>
<td><strong>Ontology</strong>:</td>
</tr>
<tr>
<td>agentive</td>
</tr>
<tr>
<td>cause</td>
</tr>
<tr>
<td>constitutive</td>
</tr>
<tr>
<td>entity</td>
</tr>
<tr>
<td>telic</td>
</tr>
<tr>
<td><strong>Event type</strong></td>
</tr>
<tr>
<td>![Input field]</td>
</tr>
<tr>
<td><strong>Pos</strong></td>
</tr>
<tr>
<td>adjactivo</td>
</tr>
</tbody>
</table>
Second part: Building and annotating RoGL
Second part: Building and annotating RoGL
Second part: Building and annotating RoGL

• To implement the generative structure and the composition rules, we have chosen the functional programming language Haskell.

• Our choice was determined by the fact that reducing expressions in lambda calculus (obviously needed in a GL implementation), evaluating a program (i.e. function) in Haskell, and composing the meaning of natural language expressions are, in a way, all the same thing.

• The most important work which still needs to be done in RoGL framework is to annotate more lexical entries. The manual annotation, although standardized and mediated by the graphical interface is notoriously time consuming especially for complex information such as those required by a generative lexicon.
Third part: On classifying coherent/incoherent short texts

• This part comprises two experiments of classification by coherence criteria of short English and Romanian texts, respectively.

• The first experiment deals with one of the new strategies adopted by spammers to send (unwanted) messages to personal e-mail accounts: encoding the real message as picture, impossible to analyze and reject by the text oriented classical filters and accompanying it by a text especially designed to surpass the filter.
Third part: On classifying coherent/incoherent short texts

- An important question for automatically categorizing texts is: are there features that can be extracted from these texts and be successfully used to categorize them? We used ratios between morphological categories from the texts as discriminant features, assuming that these ratios are not completely random in coherent text.

- We manually built a small English e-mail messages corpus comprising 110 genuine (no intervention into their text) messages: 55 negative (incoherent) and 55 positive (coherent).

- We used five supervised machine learning techniques on this corpus and let the algorithms extract important features from all 66 pos ratios.
Third part: On classifying coherent/incoherent short texts

- The first technique we used was the linear regression (Duda et al., 2001), not for its accuracy as classifier, but because, being a linear method, allows us to analyze the importance of each feature and determine the most prominent features.
- The l.o.o accuracy was of 68.18%, which we used further as baseline for next experiments.
- We ordered the 66 features (pos ratios) in decreasing order of their coefficients computed by performing regression. The top 5 features that contribute the most to the discrimination of the texts are very interesting from a linguistic point of view:
Third part: On classifying coherent/incoherent short texts

1. the ratio between modal auxiliary verbs and adverbs, representing 17.71% of all feature weights;
2. the ratio between the pre-determiner (such as all, this, such, etc) and adverbs, representing 14.6% of all feature weights;
3. the ratio between pre-determiner and conjunction, representing 9.92% of all feature weights;
4. the ratio between common nouns and conjunctions, representing 7.37% of all feature weights;
5. the ratio between modal verbs and conjunctions, representing 7.25% of all feature weights.
Third part: On classifying coherent/incoherent short texts

• These top 5 features accounted for 56.85% of data variation.
• The first ratio may be explained by the inherent strong correlation between verbs and adverbs.
• The presence of conjunction in 3 out of the top 5 ratios confirms the natural intuition that conjunction is important for text coherence.
• The presence of the pre-determiners in top 5 ratios may be related to the important role co-reference plays in the coherence of texts.
Third part: On classifying coherent/incoherent short texts

- Next, we tested two kernel methods (v support vector machine and Kernel Fisher discriminant), both with linear and polynomial kernel.
- Here are the results:

<table>
<thead>
<tr>
<th>Learning method type</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression (baseline)</td>
<td>68.18%</td>
</tr>
<tr>
<td>Linear Support Vector Classifier</td>
<td>78.18%</td>
</tr>
<tr>
<td>Quadratic Support Vector Machine</td>
<td>81.81%</td>
</tr>
<tr>
<td>Linear Kernel Fisher discriminant</td>
<td>75.46%</td>
</tr>
<tr>
<td>Polynomial Kernel Fisher discriminant</td>
<td>85.46%</td>
</tr>
</tbody>
</table>
Third part: On classifying coherent/incoherent short texts

- The best l.o.o. accuracy we obtained, i.e. 85.48% is a good accuracy because:
  - there are inherent errors, transmitted from the part of speech tagger and perhaps from the subjective human classification.
  - using only the frequencies of the parts of speech in the texts disregards other important feature for text coherence.
Conclusions of the thesis

• The unifying theme of this thesis was the concept of discourse.
• We analyzed discourse-related notions such as anaphora, quantifier scope, binding, semantic content of lexical entries or coherence, using computer science tools and methods such as continuations and machine learning techniques.
Thank you!