

EIkfEd/IDC alias BART

Yannick Versley Simone Ponzetto

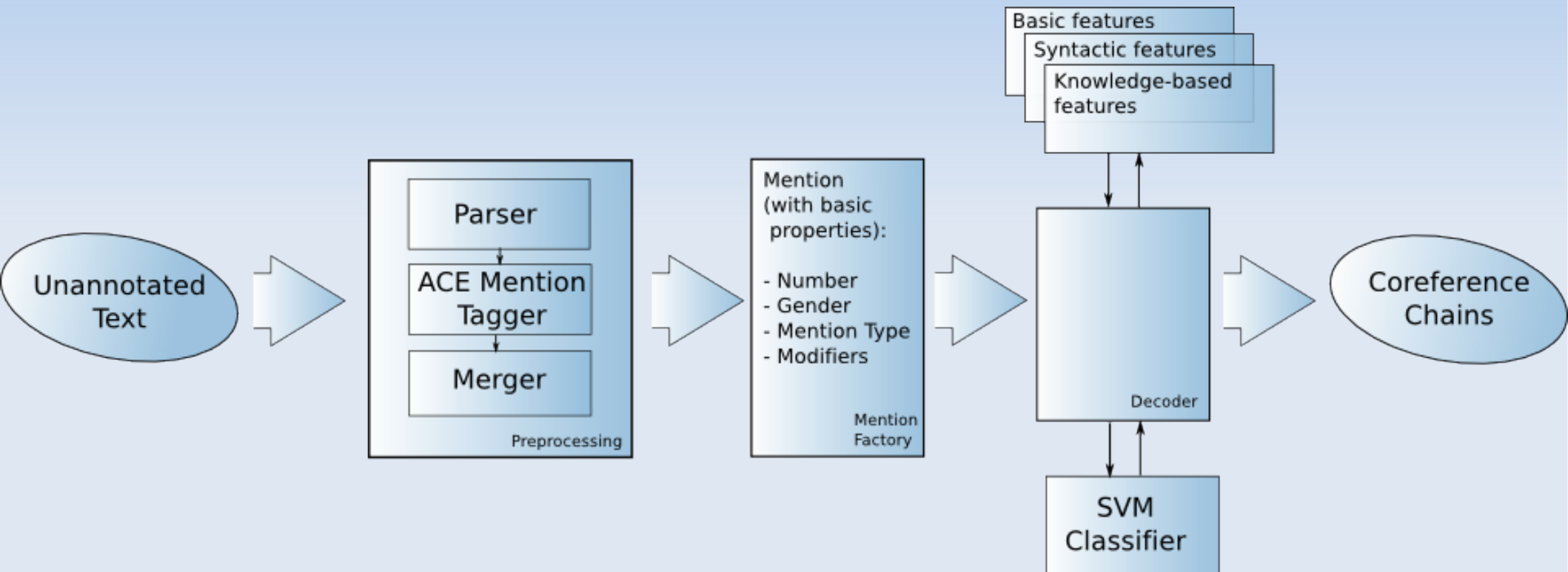
Jason Smith Vladimir Eidelman Alan Jern

Massimo Poesio Xiaofeng Yang Alessandro Moschitti

Introduction

- Modularized and revamped version of EML-R system
- Full coreference including mention detection
- Open Source release planned for December
- Useful Java-only subset (e.g. for teaching)

General data flow



Components

- Preprocessing
aggregate data in MMAX2 annotation layers
- Mention extraction
create markables from chunks/NEs
- Extract information about mentions
mention type, semantic class ...
- Encode coref into classifier decisions and extract features

Components

- Preprocessing
aggregate data in MM
- Mention extraction
create markables from chunks/NEs
- Extract information about mentions
mention type, semantic class ...
- Encode coref into classifier decisions and extract features

Modular pipeline architecture
Standoff annotation

Components

- Preprocessing
aggregate data in MMAX2 annotation layers
- Mention extraction
create markables from chunks/NEs
- Extract information about mentions
mention type, semantic class ...
- Encode coref into classifier decisions and
extract features

Feature set and learners
(and learner settings!)
described in XML file

Machine Learning Infrastructure

- WEKA Machine Learning Toolkit
 - C4.5, RIPPER, other learning modules
- SVMlight-TK
 - SVMs, different kernels (linear, polynomial, etc.)
 - Tree kernels
 - Custom kernels
- MaxEnt-based classification and ranking
 - automatic feature combination
 - ranking: find best among a set of candidates

Flexibility in Preprocessing

- Chunker (YamCha) vs. Parser (Charniak)
- Stanford NER vs. Carafe
- Mentions from Chunks/NEs vs.
Mentions from Mention Tagger (Carafe)

Resolution Algorithms

- Currently there:
 - Closest-first decoding (Soon et al., 2001)
 - Separate classifiers for pronouns/non-pronouns
 - Ranking-based resolution
 - Stacking ranking+classifier
- Wanted:
 - Classification + ranking
(cf. Ng&Cardie 2002, Yang et al 2003)
 - Global models
(Daume&Marcu 2005, Culotta et al 2007, ...)

Quantitative / Qualitative Evaluation

- MUC scoring (per document, in total)
- Link-based scores, by type of anaphora (*pronouns, appositions / copula, names, nominals*)
- Qualitative evaluation *inspect results in MMAX2*

```
MMAX2 1.12 /home/yannick/tmp/MUC-MMAX/muc6/test/ws93_022.0297.mmax [modified]
File Settings Display Tools Plugins Info ShowMLPanel
DOCID: wsj93_022_0297
DOCNO: 930504-0023 .
HL: @IBM@ appoints @ @Chrysler@ 's york@ as @finance@ chief@ --- @ @computer maker@ 's move@ signals
strategy of cuts @ in @its@ costs , asset sales @ ---- @ by @Michael W Miller@ and @Douglas Lavin@ @ staff
reporters of the wall street journal
DD: 05/04/93
wall street journal ( j ) , page 33 @ IBM@ automobiles ( aut ) , computers ( cpr )
TXT: @International Business Machines Corp.@ continued @its@ executive makeover by hiring @Jerome B. York@
an architect of the turnaround at @Chrysler Corp.@ , to become @chief financial officer@ .
@Mr. York@ , 54 years old , is a @West Point@ graduate who helped transform @Chrysler@ by slashing @costs@ and
selling billions of dollars in assets .
@His@ appointment is a strong sign that @IBM@ 's new chairman , @Louis V. Gerstner Jr.@ , plans a similar
strategy at @the wounded computer giant@ .
@Mr. Gerstner@ raced to hire @Mr. York@ after meeting @him@ for the first time just three weeks ago in @IBM@ 's
Manhattan offices .
In @his@ first month , @Mr. Gerstner@ has also brought in outsiders to run @IBM@ 's communications and
disk-drive business , and is searching for a new head of personnel .
@Mr. York@ was @executive vice president for @finance@ and a board member at @Chrysler@ , where @he@ spent
12 years in financial posts and running several @car@ and truck divisions .
@Chrysler@ did n't name a successor @yesterday@ .
```

Using different classifiers (MUC7)

Learner	Recall	Precision	F
J48	53.3	70.3	60.6
SVMlight (linear)	48.4	72.0	57.9
MaxEnt (plain)	49.4	70.4	58.0
SVMlight (polynomial d=2)	51.9	70.9	59.9
MaxEnt (combination d=2)	53.1	69.4	60.2

- J48 easiest to use, but MaxEnt/SVMlight allow for greater choice of features
- feature combination helps performance
- MaxEnt faster than SVMlight (332sec. vs. 500sec. testing time) but cannot use tree kernels

Improving Preprocessing (ACE 2002 Bnews)

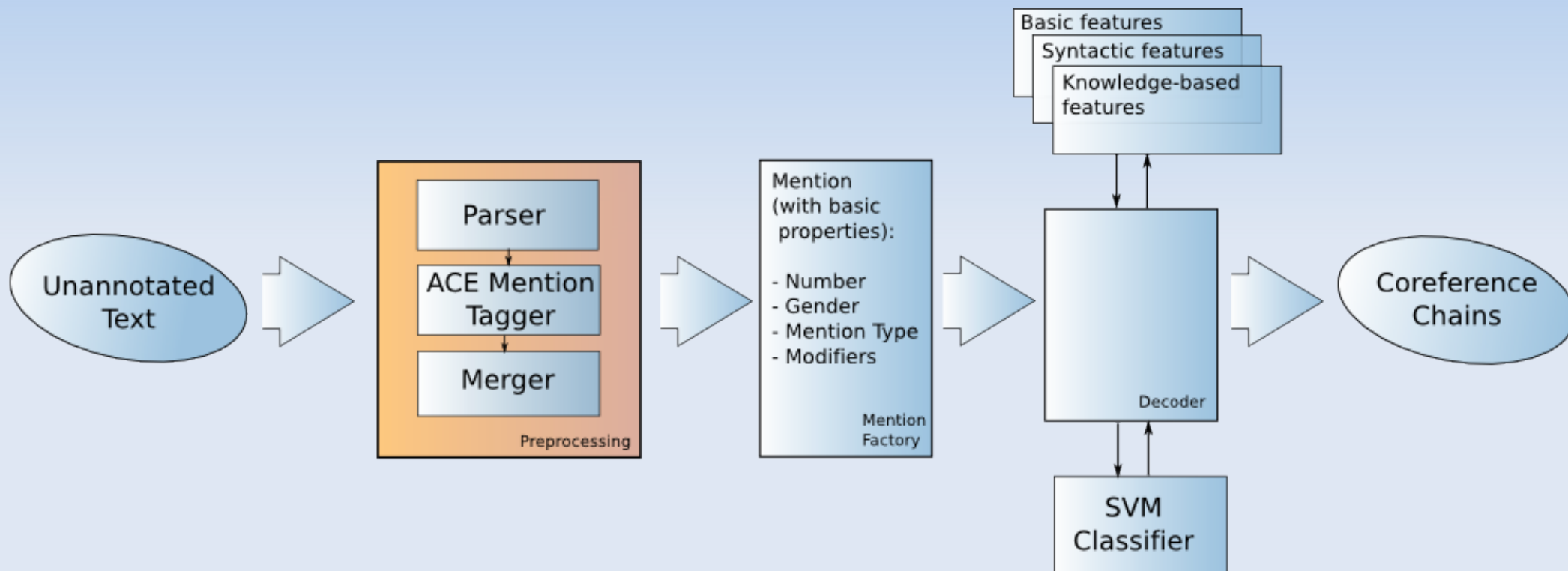
	Rec1	Prec	F
Basic Feature set	59.4	52.2	55.6
Improved preprocessing	53.7	65.7	59.1
Extended feature set	56.1	76.3	64.7

- Extended feature set: uses more syntactic information (tree kernels, salience) and world knowledge
- Eliminating preprocessing errors is important
- Depending on the corpus/annotation guidelines, corpus-specific preprocessing can be necessary.

Modifying BART

- Trying out new Feature Sets
- Adding new Feature Extractors
- Adding new Preprocessing Components
- Different Resolution Methods

Inside BART: Preprocessing



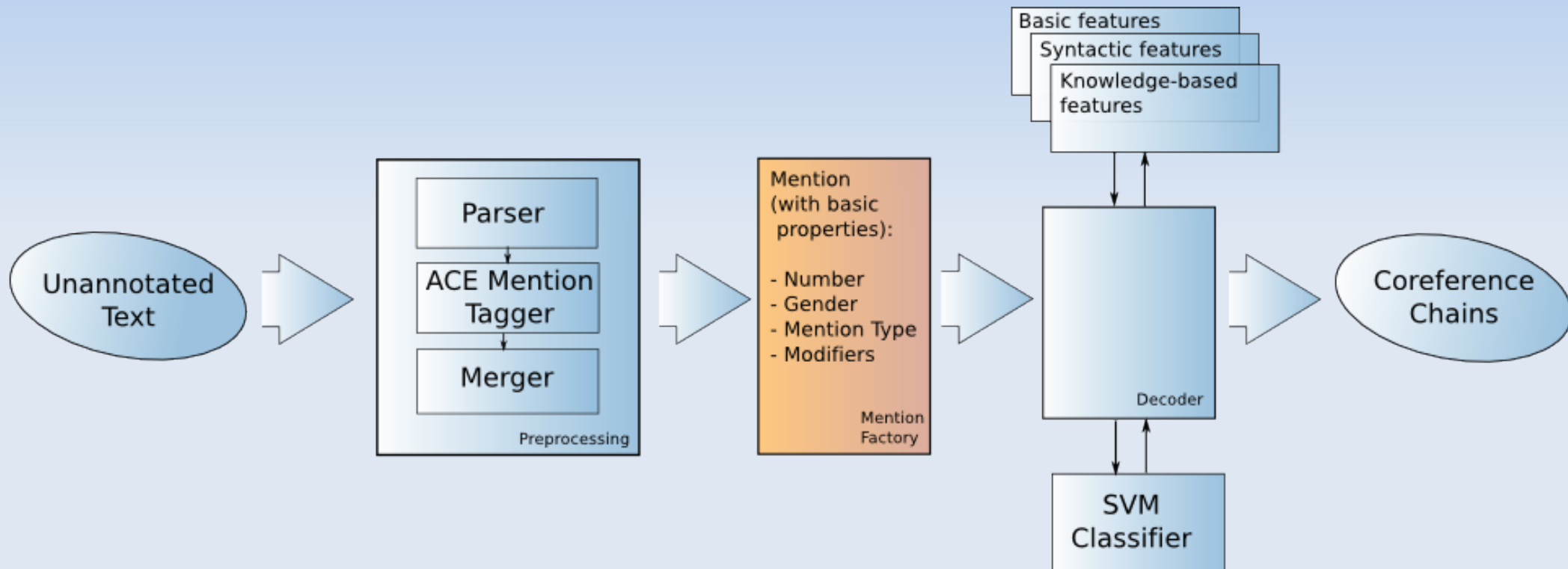
Inside BART: Preprocessing

- Documents are represented as instances of *MMAX2Discourse* (wrapped by *elkfed.mmax.Document*)
- Preprocessing information is stored on separate markable levels: Sentence, POS, Lemma, Chunk, NEs, Markable
- Chunk and NE markables are used to create markables on the Markable layer, then POS and Lemma info is added for the words

Inside BART: Preprocessing

- Pipeline components inherit from *PipelineComponent* or one of its descendants (*Parser*, *Tagger*)
- *AnnotationProcessor* or *TrainerProcessor* invokes the pipeline, which in turn calls the preprocessing components.

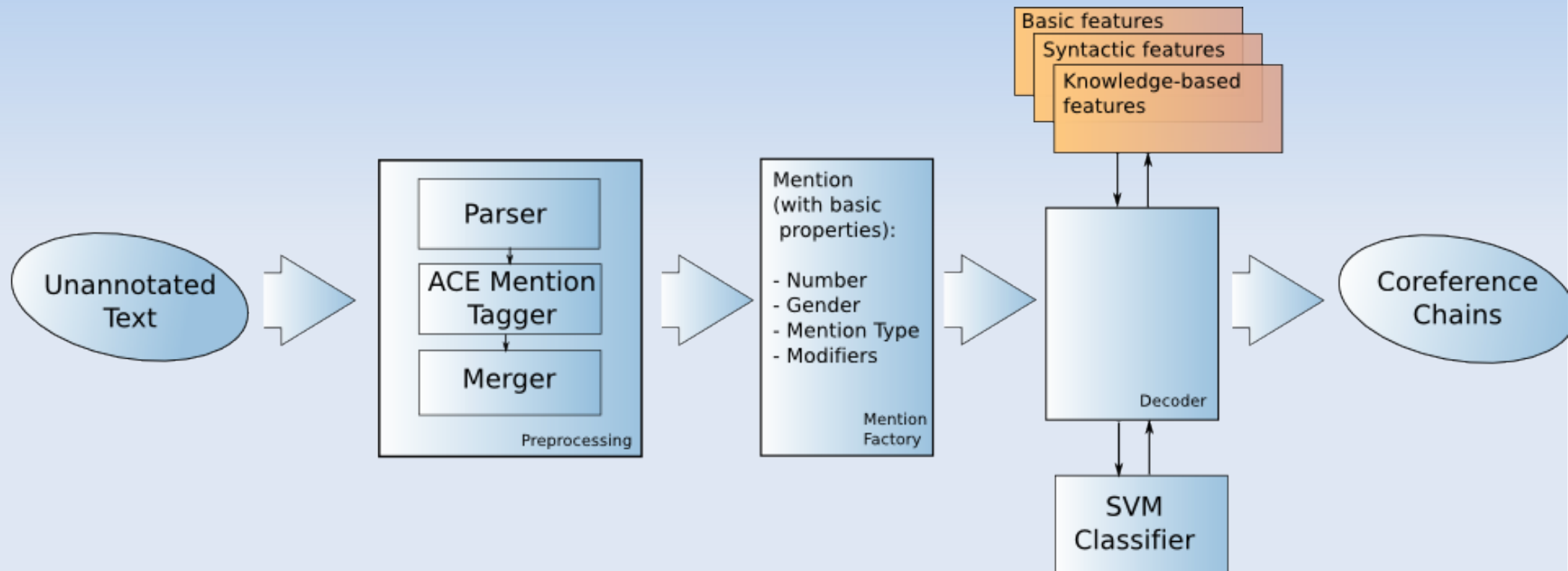
Inside BART: Mentions



Inside BART: Mentions

- a *MentionFactory* creates instances of the *Mention* class by
 - taking information from the Markable layer (each Markable markable becomes one mention)
 - taking parses from the Parse annotation layer
 - linking Mentions to their Utterance
- *Mention* is responsible for
 - keeping track of important information about the markable (gender, number, NE/not NE)
 - linking to a node in the parse tree (no node will be present if we use a chunker)

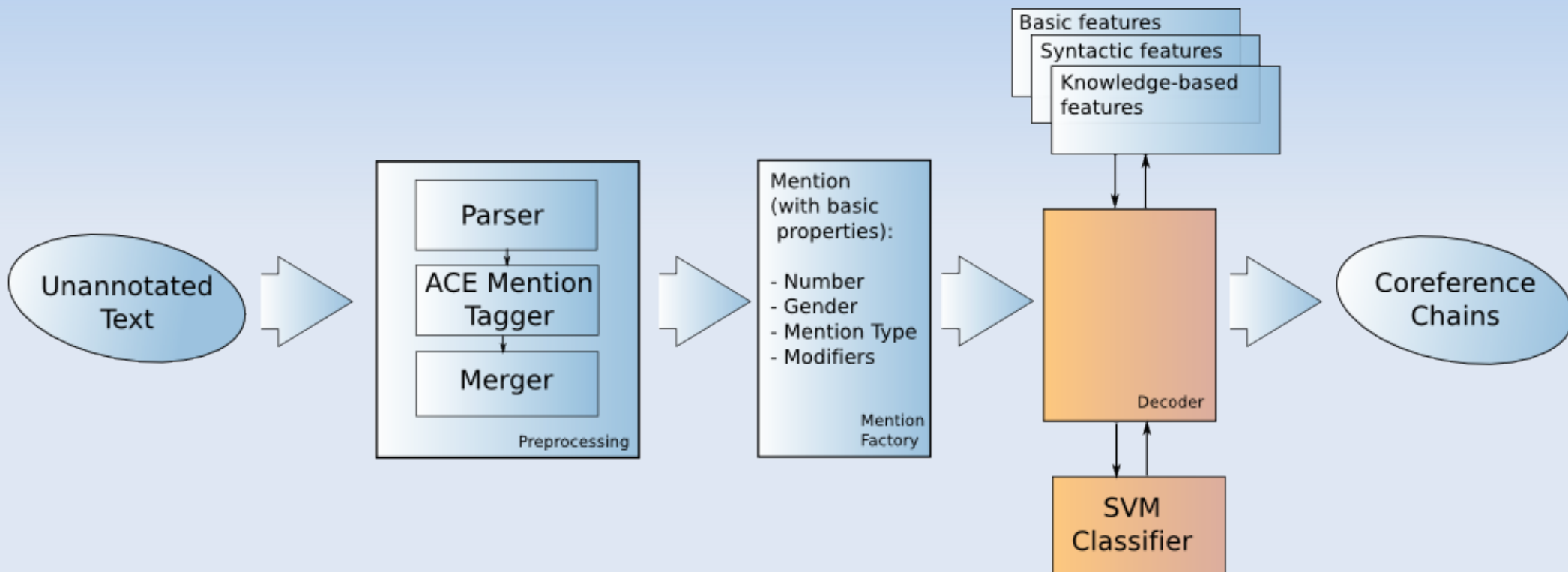
Inside BART: Features



Inside BART: Features

- *PairInstance*
 - contains references to antecedent and anaphor *Mentions*
 - contains features of that pair (string match, alias, ...)
- *PairFeatureExtractor*
 - extracts one or several features
 - describeFeatures: give list of features and their type (*FeatureDescription*)
 - extractFeatures: takes a *PairInstance* and adds the features

Inside BART: Learning



Inside BART: Learning

- Learning is done by *CorefTrainer* subclasses, Annotation by *CorefDecoder*.
- *CorefTrainer* gets handed the mentions in the document and writes training examples to one or more *InstanceWriter* instances.
- *CorefDecoder* constructs testing examples and hands them to an *OfflineClassifier* constructed from the training examples.
It returns a partition of *Mentions*

XML Configuration File

```
<coref-experiment>
<system type="soon">
  <classifiers>
    <classifier type="weka" model="idc0"
      learner="weka.classifiers.trees.J48"
      options="" />
  </classifiers>
  <extractors>
    <!-- feature extractors -->
    <extractor name="FE_MentionType" />
    ...
    <extractor name="FE_SentenceDistance" />
  </extractors>
</system>
</coref-experiment>
```