DFKI System Combination with Sentence Ranking

@ ML4HMT-2011

Eleftherios Avramidis
18.11.2011, Barcelona

Overview

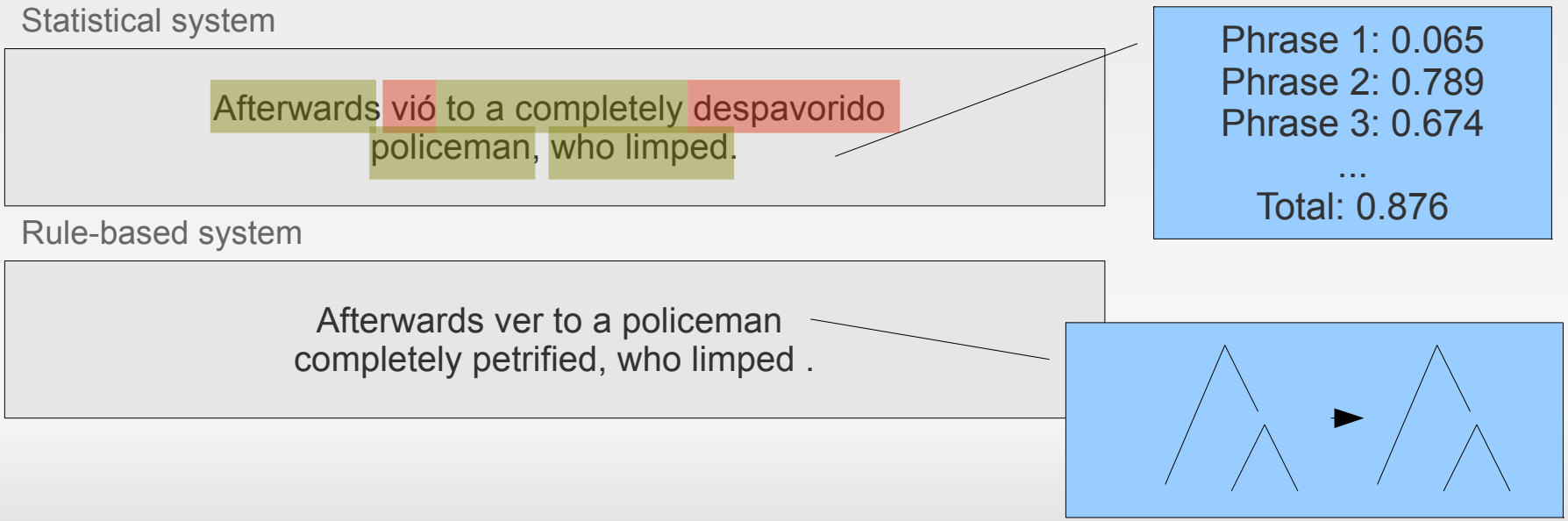
1. Re-formulate the problem in a ML-friendly way
2. Investigate and extract features
3. Machine Learning algorithms and performance
4. Discussion of the results

Challenge

- **Input:**
  - Development corpus with 1,000 sentences
  - Test corpus with 1,000 sentences
  - 5 “annotated” system outputs per sentence
- **Goal:** Use machine learning in order to combine system outputs in an optimal way
- **Idea:**
  - Learn from development corpus and apply to test corpus
  - Try to use system meta-data as features

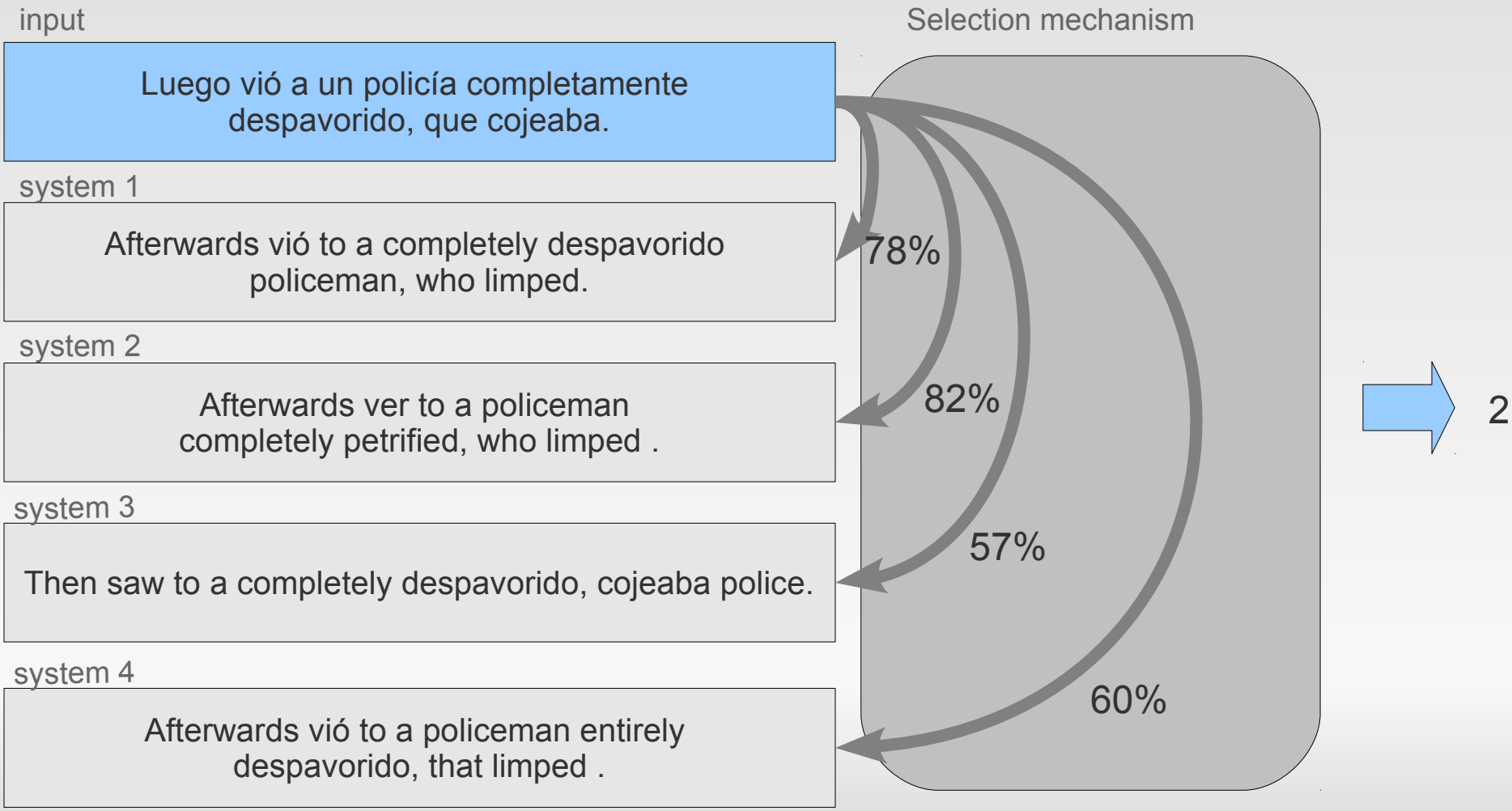
1. Reformulation of the problem: sentence level

- 1,000 sentences, translated by 5 systems each
- Rich annotation – heterogeneous and overlapping metadata
- **Goal**: find a common basis between all outputs, so that feature vectors for ML make sense
- **Solution**: restrict granularity to the **sentence level**



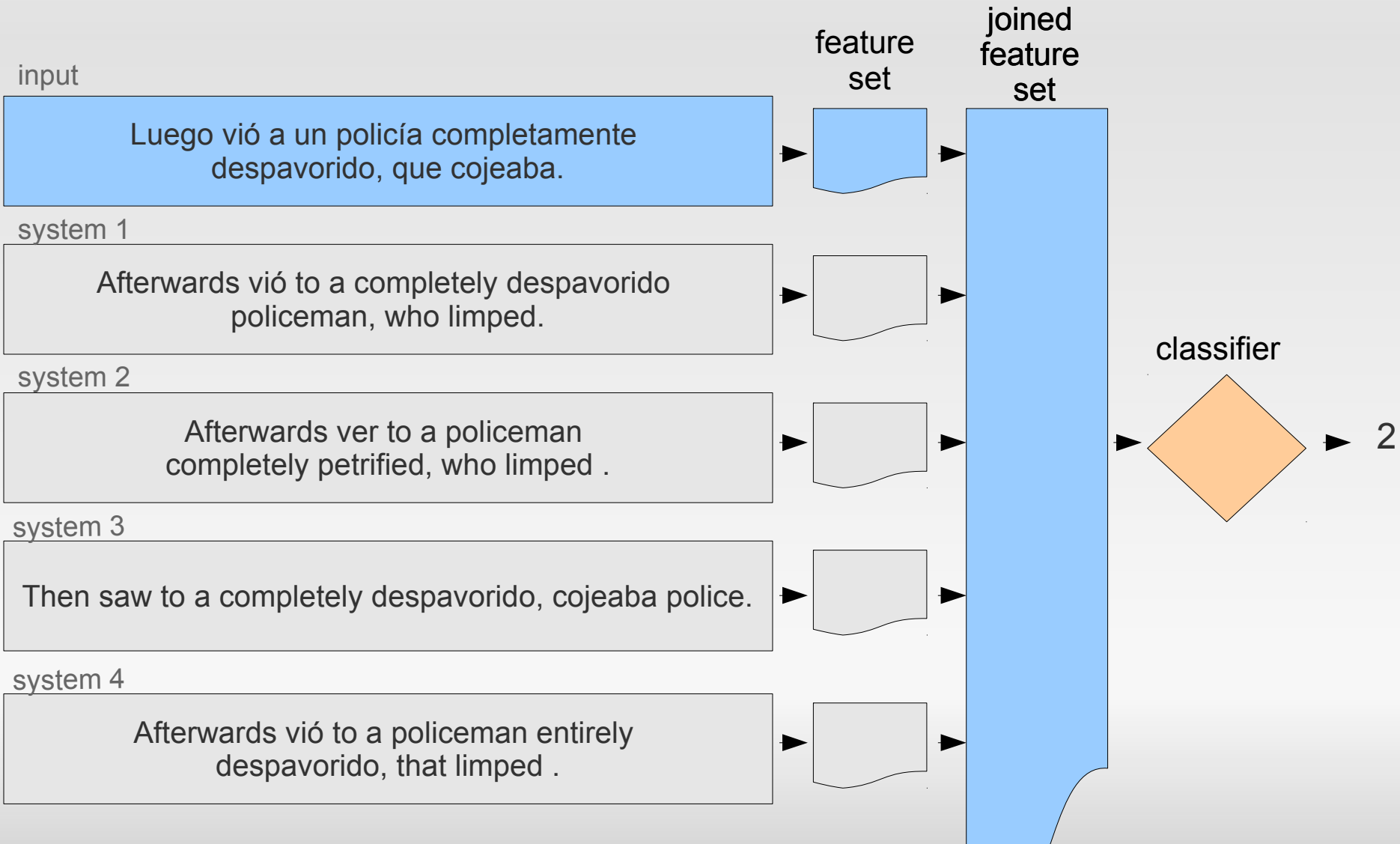
1. Reformulation of the problem: quality estimation

- **Goal**: empirical selection mechanism, able to learn about the quality of the outputs on the fly and choose accordingly.



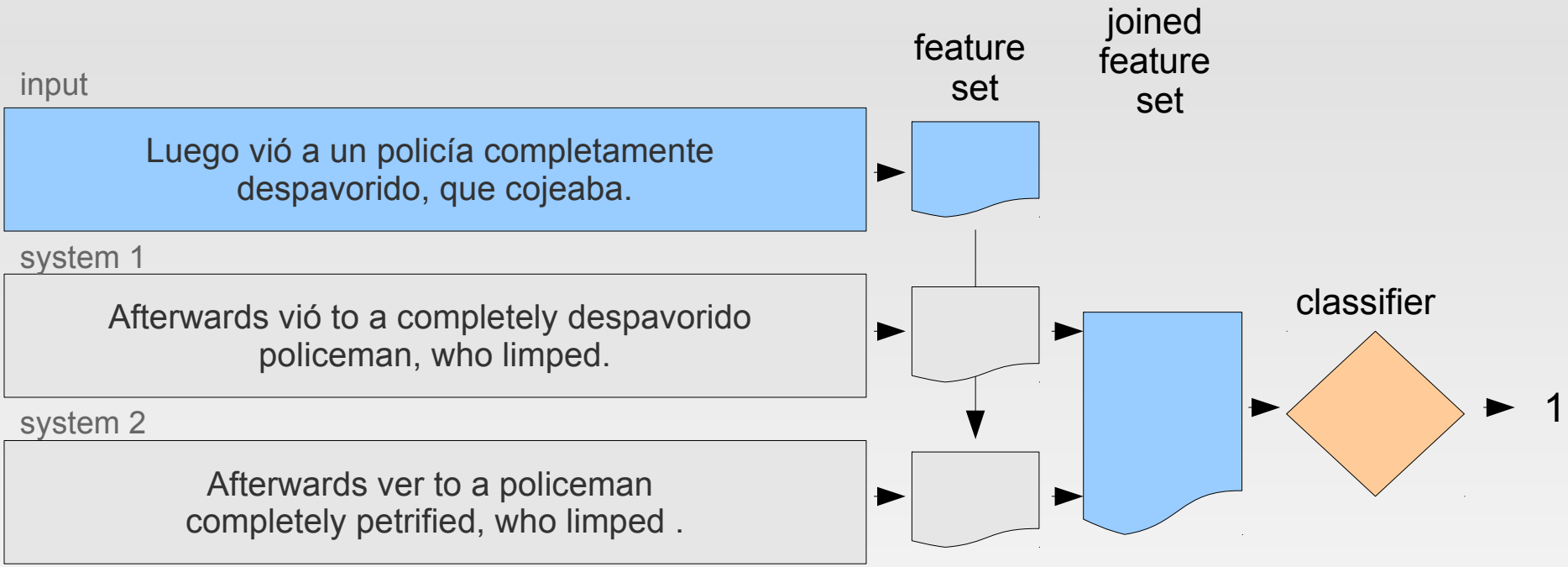
1. Reformulation of the problem: everything in one decision?

- **Heavy approach**: Learn everything at once



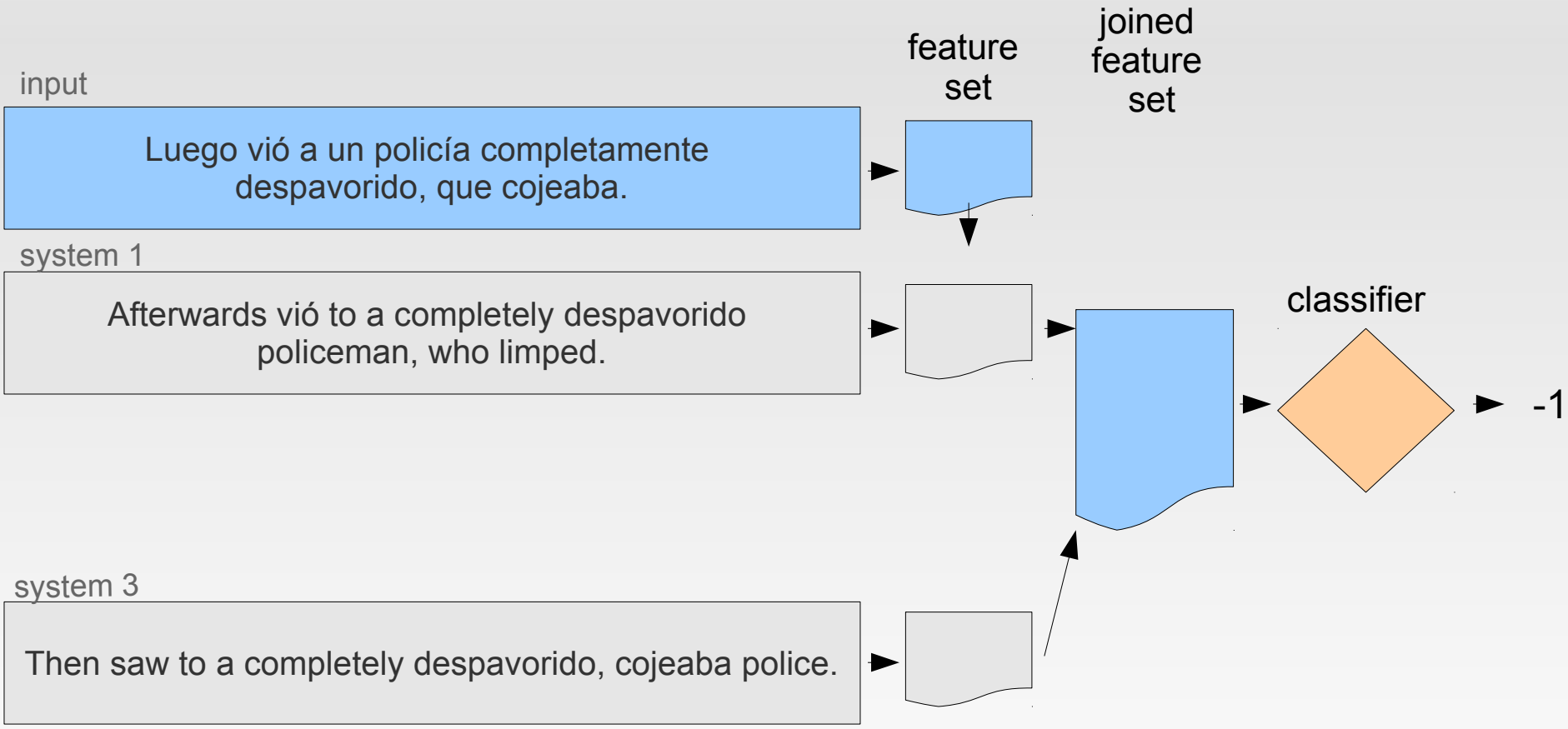
1. Reformulation of the problem: pairwise decisions

- Compare two sentence outputs at a time and decide which one is better



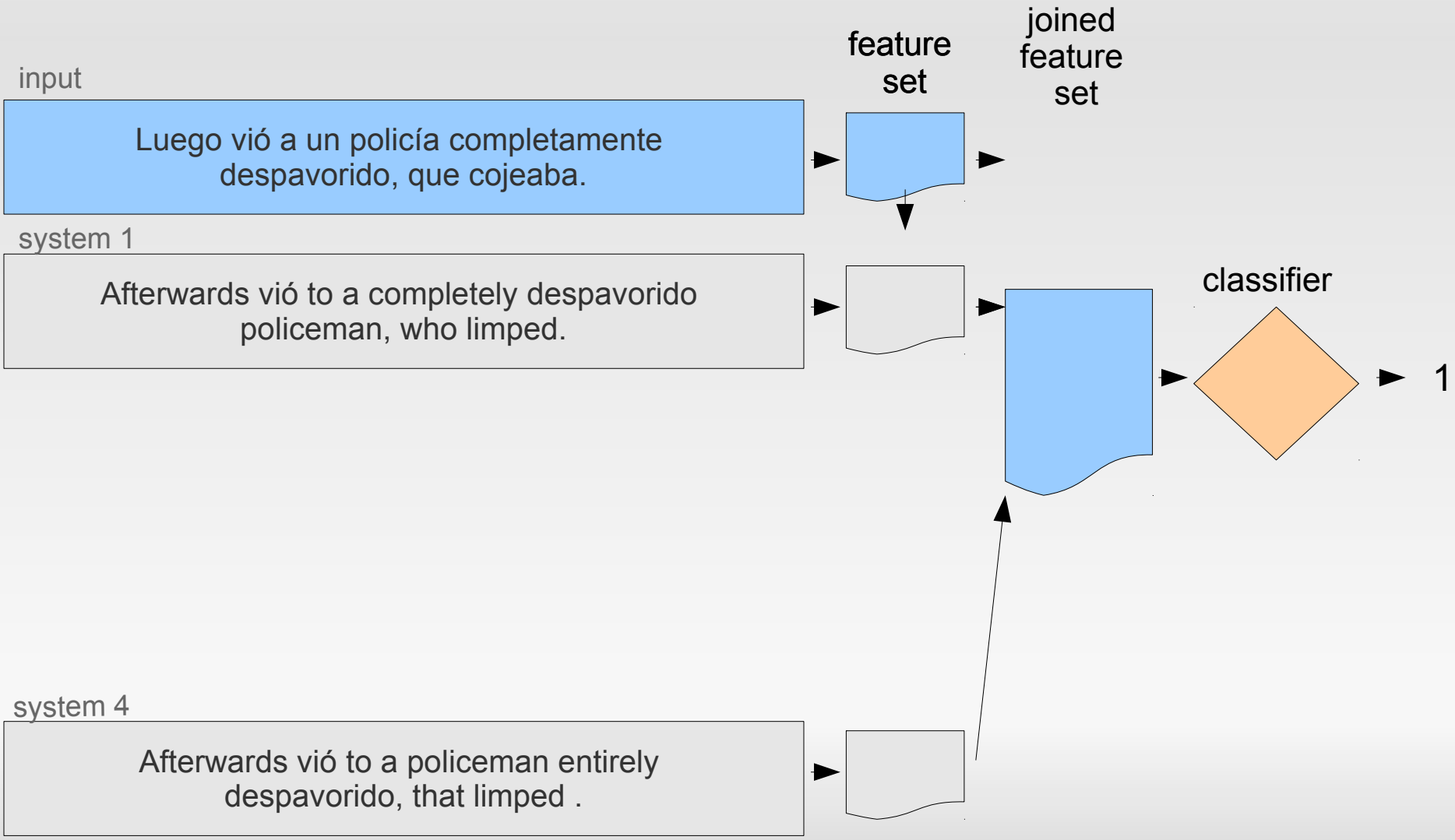
1. Reformulation of the problem: pairwise decisions

- Compare two sentence outputs at a time and decide which one is better



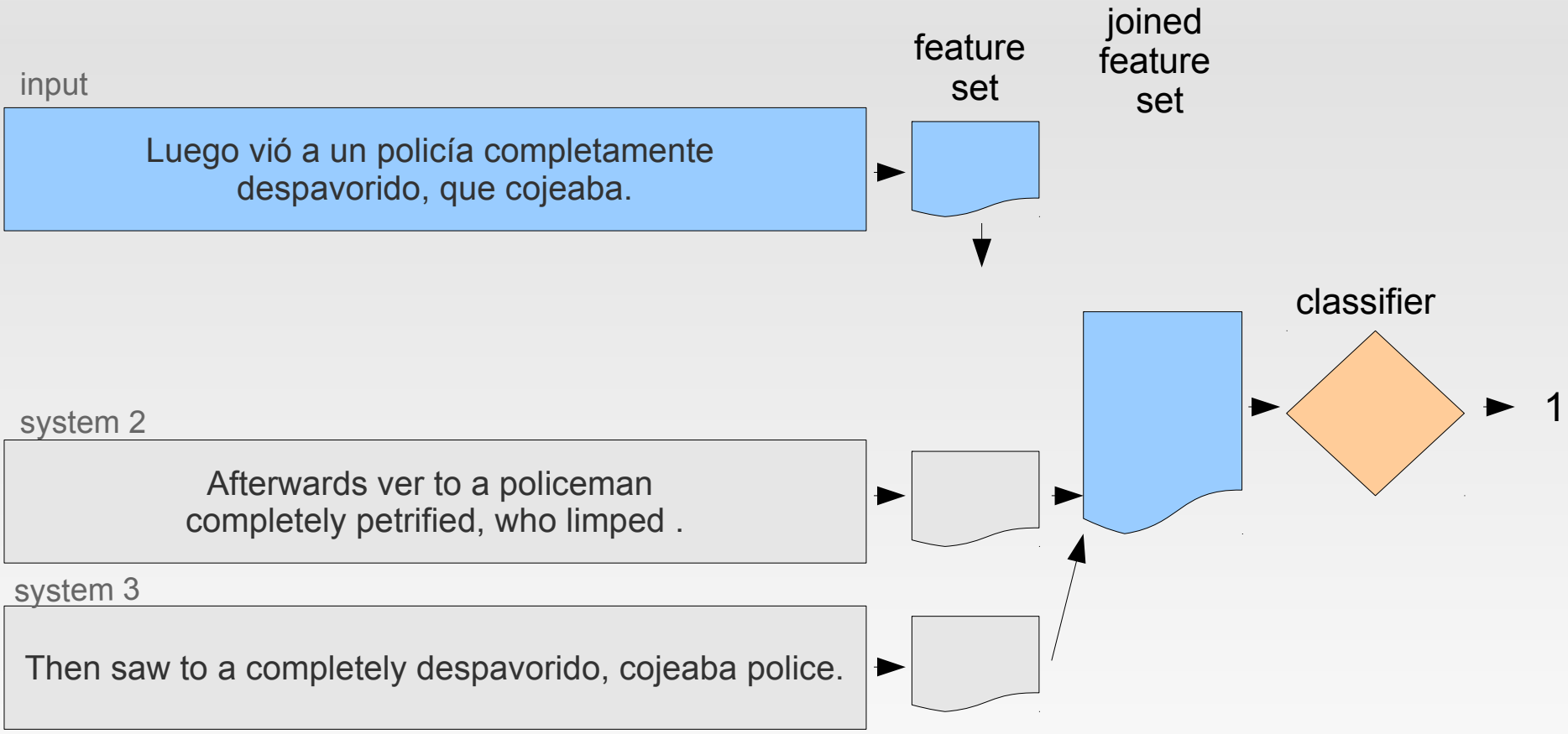
1. Reformulation of the problem: everything in one decision?

- Compare two sentence outputs at a time and decide which one is better

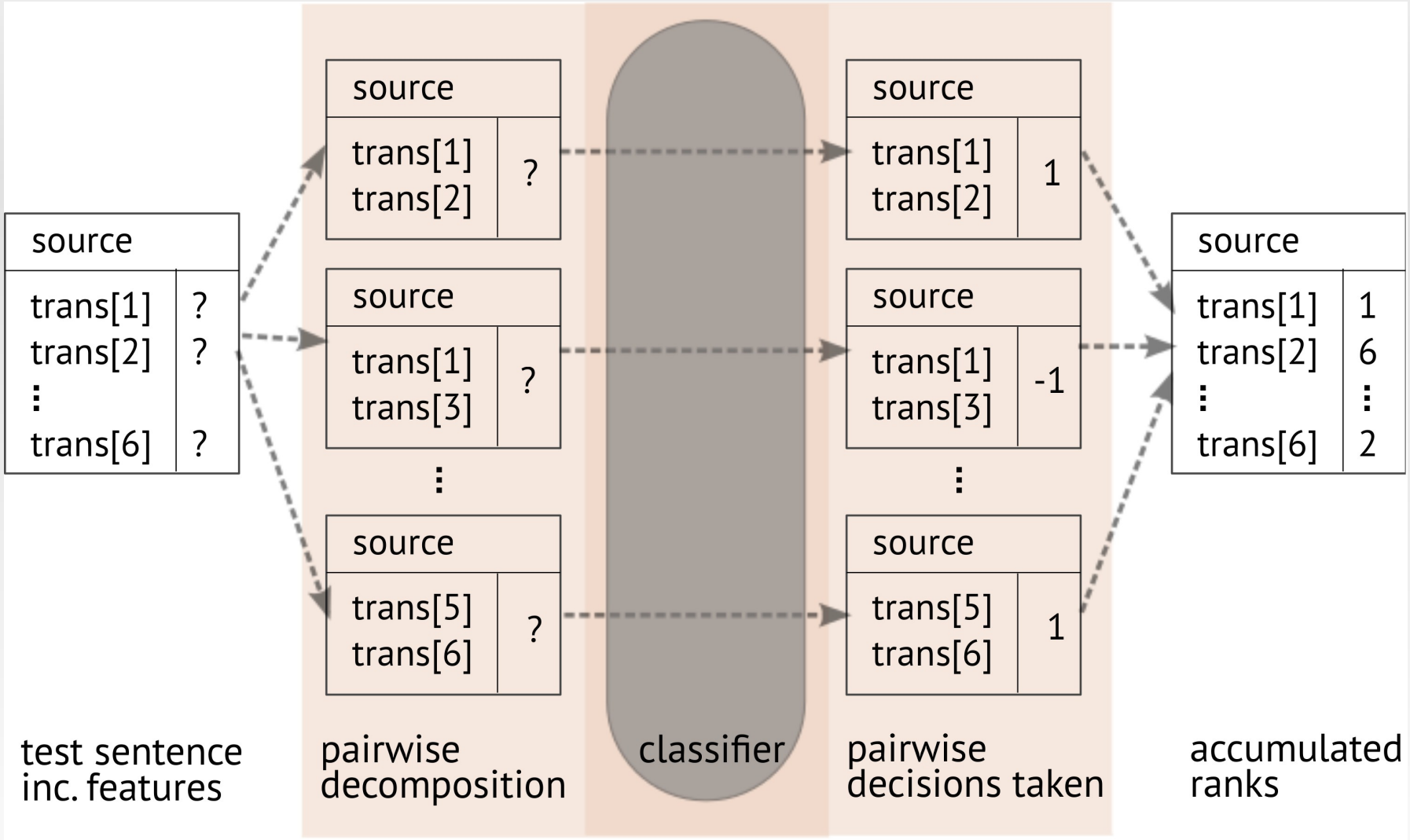


1. Reformulation of the problem: pairwise decisions

- Compare two sentence outputs at a time and decide which one is better



1. Reformulation of the problem: pairwise decisions



The process of Machine Ranking, performed through pairwise comparisons for 6 system outputs

1. Reformulation of the problem: pairwise decisions

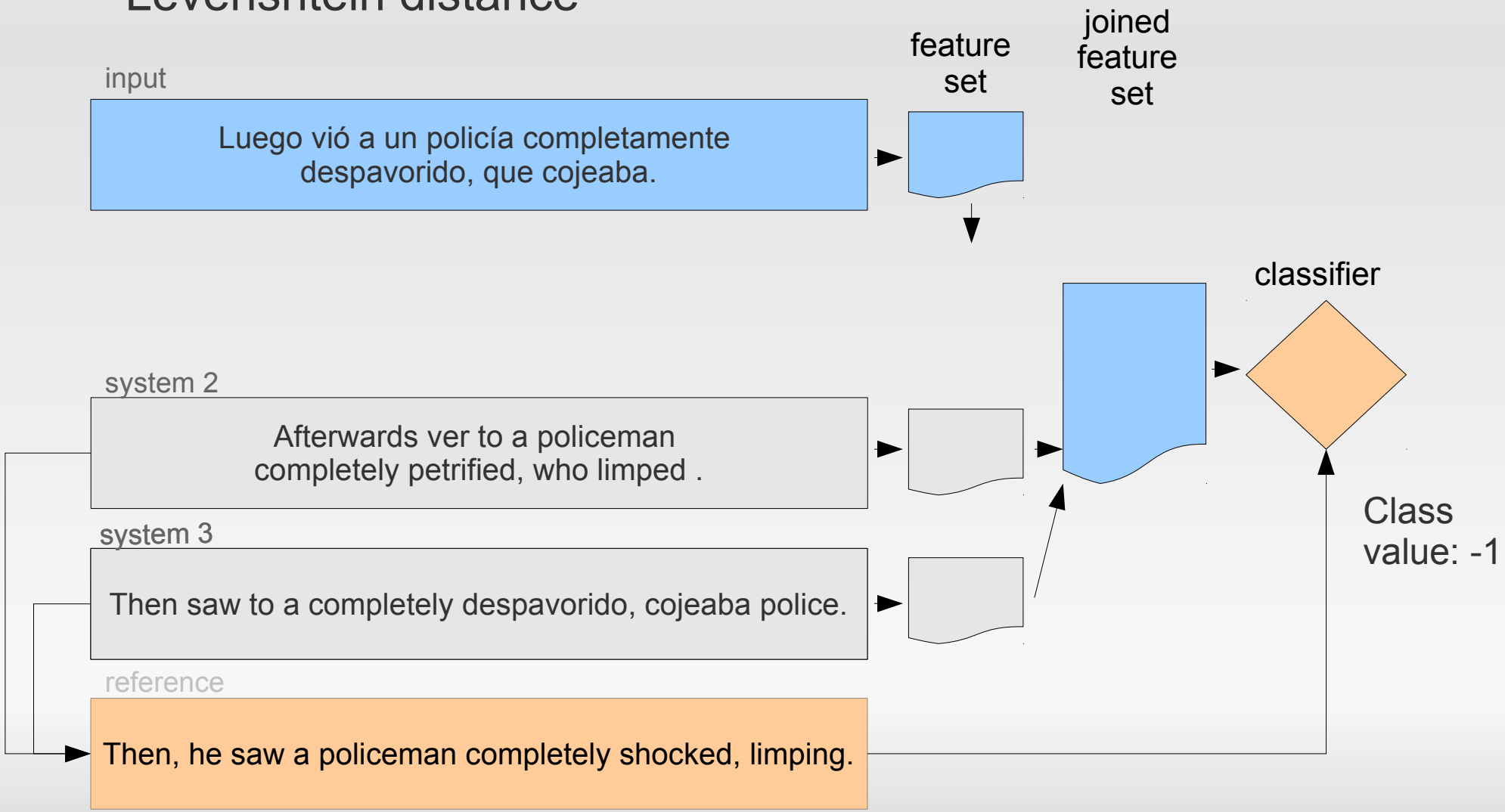
- System chooses the output that won the most pairwise comparisons
- 1,000 sentences result into 17,000 training instances with binary classes – less sparseness
- Simple question posed to the ML:

  Which of the two outputs is better?

- Independent of the order of the system outputs

1. Reformulation of the problem: supervised learning

- Train binary classifier with class labels derived by word-level Levenshtein distance



2a. Extracting sentence features

Joshua:

- Sentence probability
- Phrase count
- Decoding algorithm statistics:
  - Pre-pruned nodes
  - Added nodes
  - Merged nodes
  - Fuzzy matches
- Feature scores for every decoding step (TM, LM):
  - Average, variance, standard deviation

MaTrEx:

- Sentence probability
- Phrase count
- Feature scores for every decoding step (transition probability, future cost estimate):
  - Average, variance, standard deviation

2a. Extracting sentence features

Lucy:

- Count of tags in analysis/transfer tree nodes
  - Tags that indicate that phrasal analysis took place

Others:

- External tools:
  - Language model probabilities
    - Bigram
    - Trigram
    - 5-gram
  - PCFG parse probability

2b. Feature selection

feature	Inf.gain	Gain ratio	Gini index
Lucy phrasal analysis	0.181	0.092	0.059
Joshua total probability	0.100	0.050	0.030
External 5gram score	0.000	0.037	0.000
MaTrEx std deviation of future cost	0.058	0.029	0.019
MaTrEx std deviation of phrase prob.	0.058	0.029	0.019
Joshua/MaTrEx phrase count	0.012	0.005	0.004

feature	ReliefF
Joshua total probability	0.064
Lucy phrasal analysis	0.023
MaTrEx total probability	0.012
Joshua merged nodes	0.011
Joshua word penalty variance	0.010

3 Classification

- Binary classifiers trained:
  - SVM
  - Naïve Bayes
  - Linear

- For Naïve Bayes and Linear classifier:
  - Feature selection
  - Imputation of missing values
    (the most frequent value is used for imputation)

3 Learning algorithms and results

classifier	pairwise accuracy	segment Kendall correlation (τ)	select-best accuracy
SVM	0.52	**0.52**	0.53
Bayes	0.63	0.43	0.54
Linear	0.51	0.25	0.50

- Classifiers managed to provide the best solution right away in 50-54% of the cases. (the probability of random selection out of the five alternatives would be 20%)
- Manual evaluation (SVM): the classifier comes to a level of uncertainty concerning the two best ranked sentences
- The classifier built with SVM gives the best average sentence-level correlation.
  - only 6% of the sentences had a negative tau coefficient.
- Tau correlation given in this task is much higher than the ones achieved by evaluation metrics in WMT (but there human rankings)

Overall performance

System	BLEU	NIST	METEOR	PER	WER
Joshua	19.68	6.39	50.22	47.31	62.37
Lucy	23.37	6.38	57.32	49.23	64.78
Metis	12.62	4.56	40.73	63.05	77.62
Apertium	22.30	6.21	55.45	50.21	64.91
MaTrEx	23.14	6.71	54.13	45.19	60.66
DFKI-A	**23.54**	**6.59**	**54.30**	**61.31**	**46.17 ***

Table 1: Automatic scores for combined test output

System	Annotator #1	Annotator #2	Annotator #3	Overall
DCU	2.44	2.61	2.51	2.52
DFKI-A	**2.50**	**2.47**	**2.48**	**2.48**
DFKI-B	2.06	2.13	1.97	2.05
LIUM	2.89	2.79	2.93	2.87

Table 2: Human rankings for all systems

Further work

- Better features from all systems
- Correlation feature selection
- Eliminate ties when selecting best sentence (more fine-grained score/confidence generation)
- Obtain class values with a more state-of-the-art sentence-level metric OR
- Train classifiers given human annotations

Thanks!

- Many thanks to Lukas Poustka for preprocessing and training the Spanish grammar.

- This work has been developed within the TaraXÜ project financed by TSB Technologiestiftung Berlin – Zukunftsfonds Berlin, co-financed by the European Union – European fund for regional development.