THE IMPORTANCE OF SYNTACTIC PARSING AND INFERENCE IN SEMANTIC ROLE LABELING

(PUNYAKANOK ET AL. 2008)

Summer Semester
June 27th 2011

Peter Michael Stahl

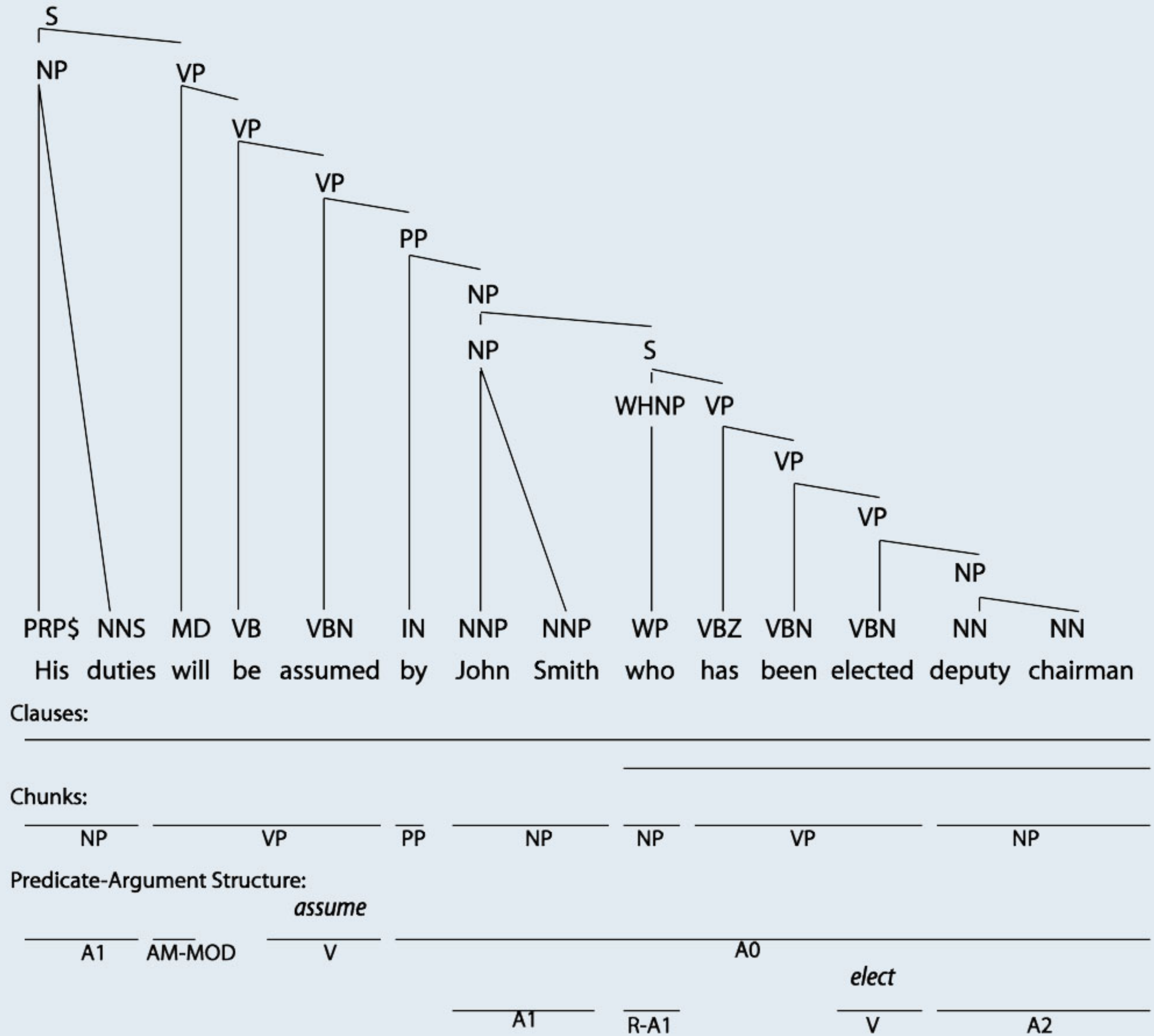
Seminar: Recent Developments in Computational Semantics

Outline

- SRL system architecture
  - Pruning
  - Argument identification
  - Argument classification
  - Inference
- What is Integer Linear Programming (ILP)?
- Importance of Syntactic Parsing
  - full syntactic parsing
  - shallow syntactic parsing
- Conclusion

3

SRL System Architecture


S
NP
VP
VP
VP
PP
NP
NP
S
WHNP
VP
VP
VP
NP
PRP$ NNS MD VB VBN IN NNP NNP WP VBZ VBN VBN NN NN
His duties will be assumed by John Smith who has been elected deputy chairman
Clauses:
Chunks:
NP
VP
PP
NP
NP
VP
NP
Predicate-Argument Structure:
assume
A1
AM-MOD
V
A0
elect
A1
R-A1
V
A2

Two-step design

- 1) System is trained to identify argument candidates for a given verb predicate
- 2) Argument candidates are classified into their types
- Additional steps:
  - Pruning of obvious non-candidates before step 1
  - Use post-processing inference to fix inconsistent predictions after step 2
- System uses PropBank Annotation Scheme

Pruning step

- filter out simple constituents that are unlikely to be arguments
- recursive process starting from the target verb
- returns siblings of verb as candidates
- then it moves to parent of the verb and collects siblings again
- process continues until it reaches the root
- exploits heuristic rules introduced by Xue and Palmer (2004)

Argument identification step

- binary classification
- full syntactic parsing:
  - train and apply classifiers on constituents supplied by pruning stage
- shallow syntactic parsing:
  - no pruning stage → consider all possible subsequences (i.e. consecutive words) as potential candidates
  - two classifiers: One to predict beginnings of a potential candidate, the second to predict the ends
  - predictions are combined

Features of classifier

- same features as in Gildea & Jurafsky (2002)
- some additional features which are especially useful for systems without full parse tree information, for example:
  - two word before and after constituent, together with their POS tags
  - length of constituent, measured in words and chunks
  - chunk pattern: encodes sequence of chunks from constituent to predicate
  - chunk pattern length: counts number of chunks in chunk pattern feature

Argument classification step

- multi-class classifier used to predict semantic types or label argument as *null* to discard it
- same features as in identification step
- for details of the classifier see (Roth 1998) and (Carlson et al. 1999)

Inference step

- previous decisions were made for each argument independently, ignoring global information across arguments
- purpose: incorporate such information here, including linguistic and structural knowledge, e.g.:
  - *arguments do not overlap*
  - *each verb takes at most one argument of each type*
- → useful to resolve inconsistencies of argument classification

Inference step

- process of finding the best valid semantic labels which satisfy certain constraints
- **input**: the argument classifier's confidence scores for each type of argument, along with a list of constraints
- **output**: optimal solution that maximizes linear sum of confidence scores, subject to constraints that encode domain knowledge
- accomplished through integer linear programming

Constraints over Argument Labeling

- $S^{1:M}$: set of arguments, indexed from 1 to M
- P^M: set of labels
- the indexed set of arguments can take a set of labels $c^{1:M} \in P^M$
- classifiers return $score(S^i = c^i)$
- → likelihood of argument S^i being labeled c^i, given a sentence
- goal: maximize the overall score of arguments

$$\hat{c}^{1:M} = \underset{c^{1:M} \in \mathcal{F}(\mathcal{P}^M)}{\operatorname{argmax}} \sum_{i=1}^{M} \operatorname{score}(S^i = c^i)$$

Constraints over Argument Labeling

find the set of labels for which the sum of the scores of each argument is maximized

can be thought of as if the solution space is limited through a filter function *F* which eliminates many labelings from consideration

$$\hat{c}^{1:M} = \underset{c^{1:M} \in \mathcal{F}(\mathcal{P}^M)}{\operatorname{argmax}} \sum_{i=1}^{M} \operatorname{score}(S^i = c^i)$$

Constraints over Argument Labeling

the filter function uses several constraints, e.g.:

- arguments cannot overlap with the predicate
- if a predicate is outside a clause, its arguments cannot be embedded in that clause

$$\hat{c}^{1:M} = \operatorname*{argmax}_{c^{1:M} \in \mathcal{F}(\mathcal{P}^M)} \sum_{i=1}^{M} \text{score}(S^i = c^i)$$

15

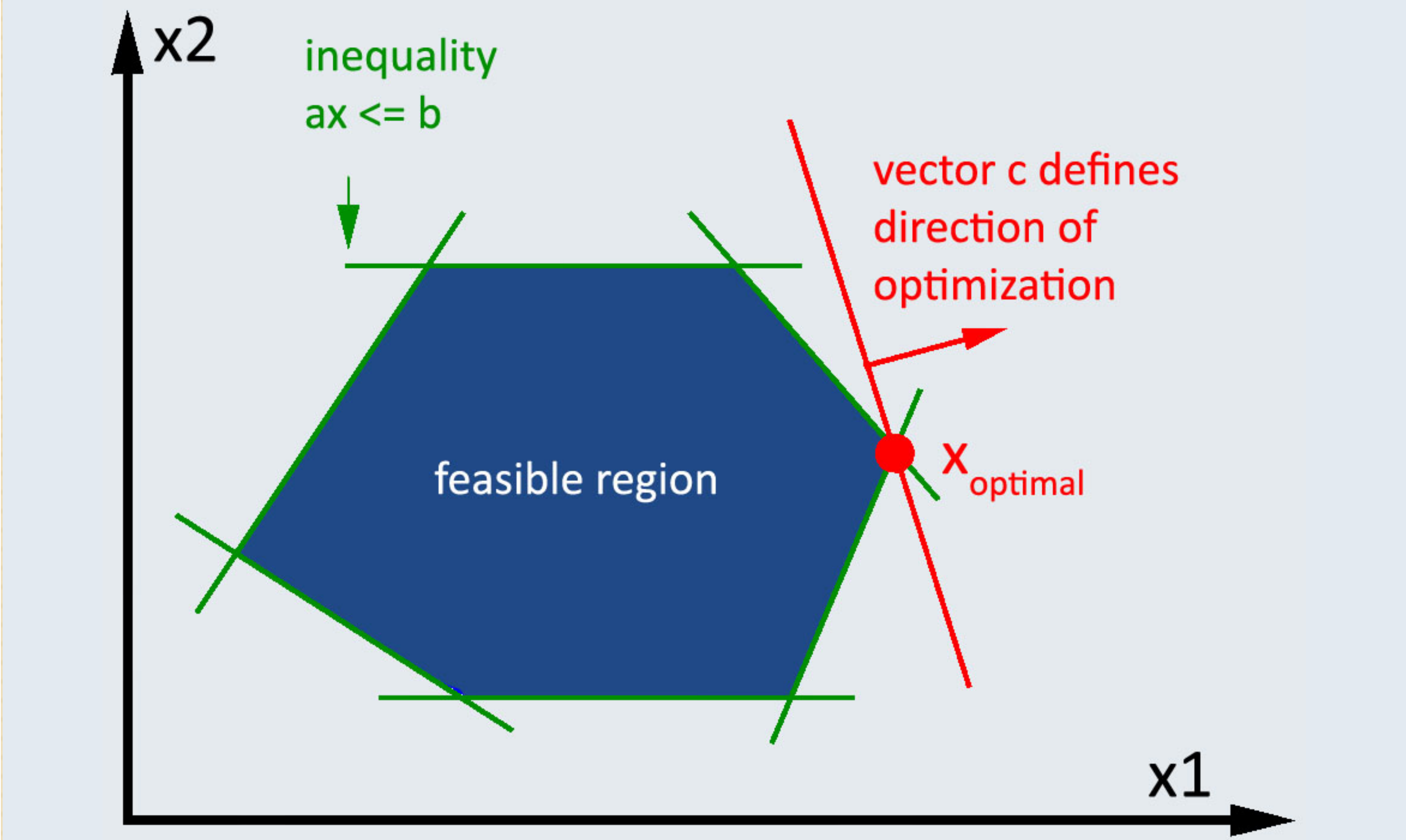
Constraints over Argument Labeling

problem: computation takes a very long time, but can be optimized by using an integer linear programming resolver

Integer Linear Programming

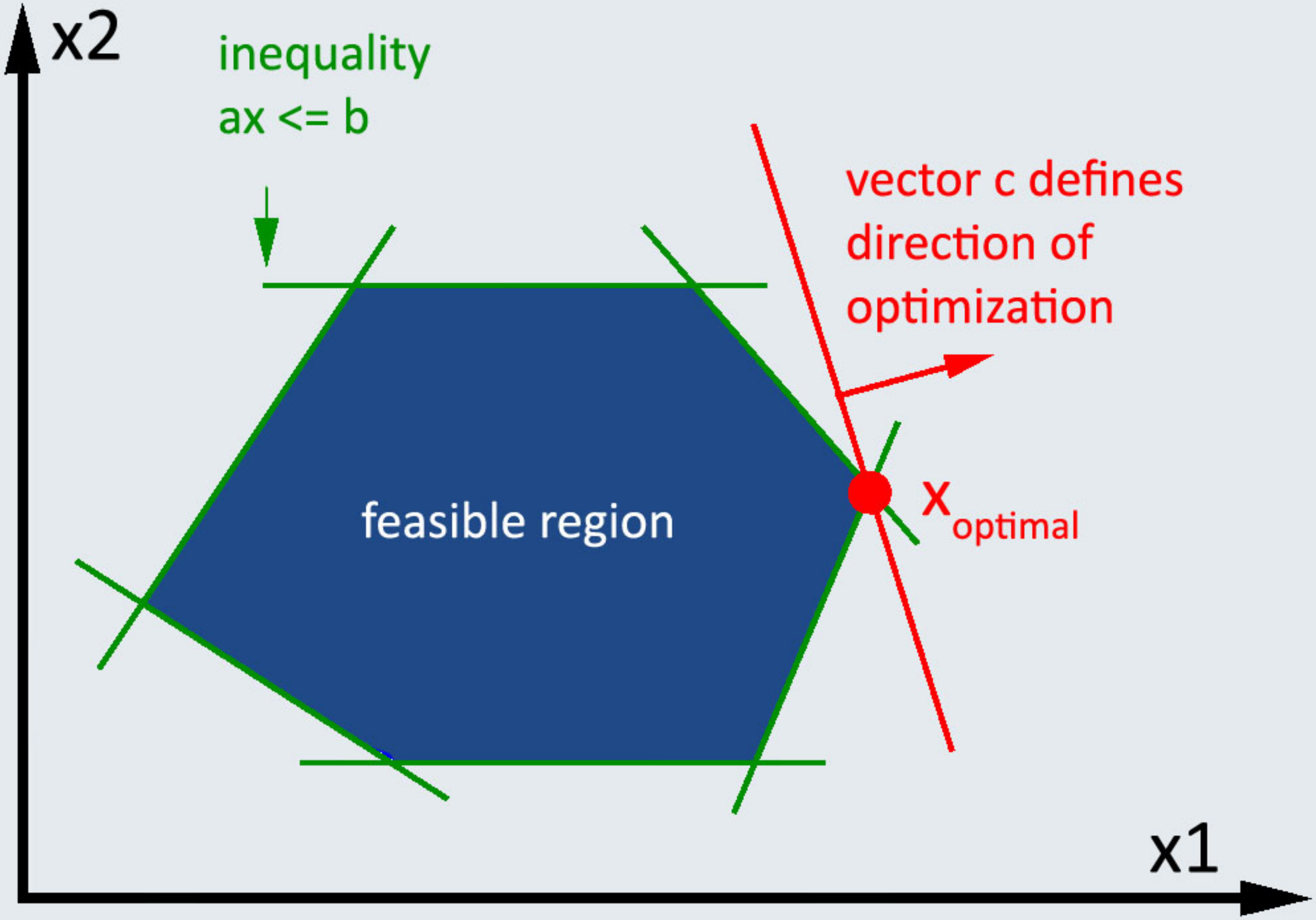
What is ILP?

- ILP is a special case of linear programming
- also called linear optimization
- mathematical method for achieving the best outcome in a mathematical model for some list of requirements represented as linear relationships
- → maximize an objective function depending on linear (in)equality constraints
- if unknown variables are all required to be integers, then the task is called ILP



Geometrical interpretation of ILP

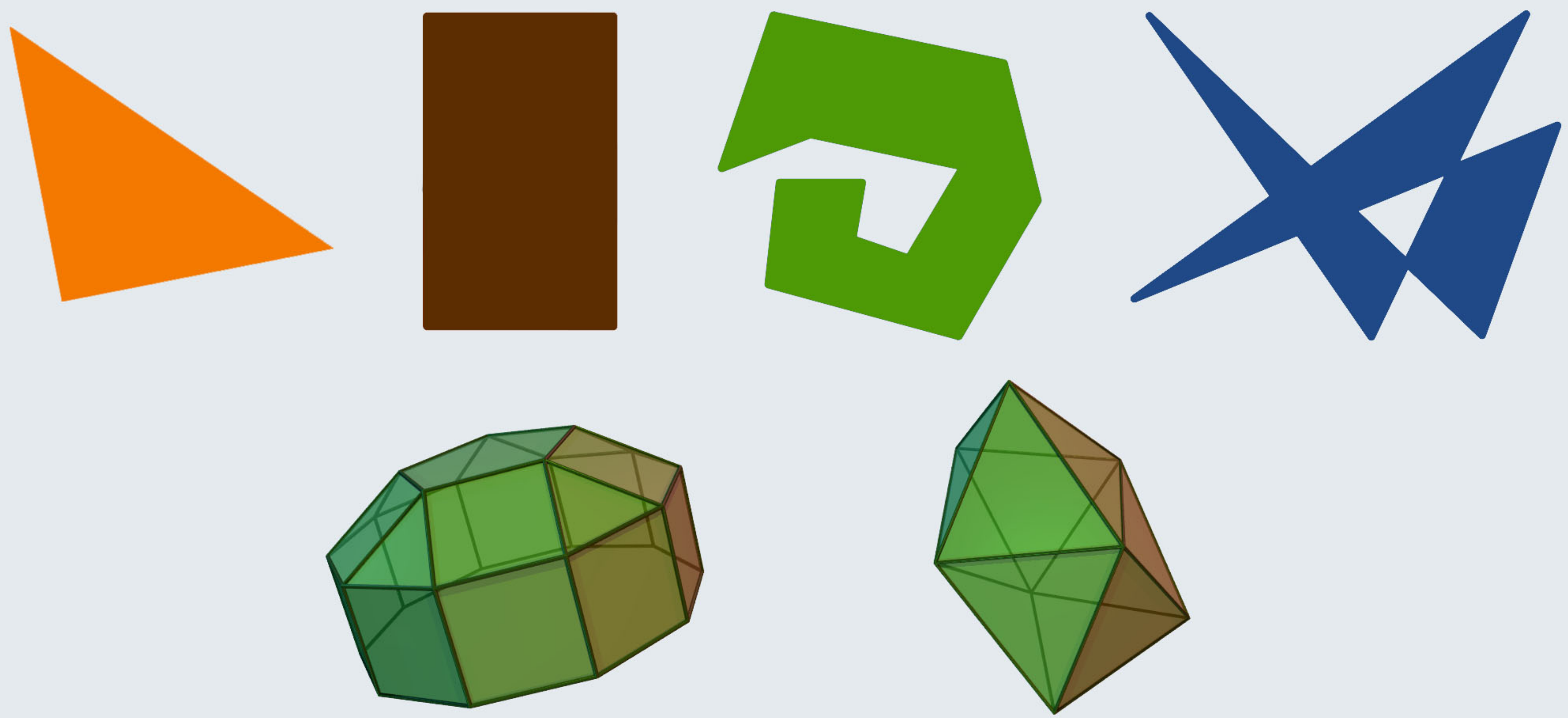
- set {x | ax = b}: all points x which satisfy ax=b (a and b are constants)
- all these points define a hyperplane in n-dimensional space
- all points which satisfy ax <= b lie on one side of this hyperplane
- each constraint divides the space into two parts; points on one side are feasible, those on the other side are not



19

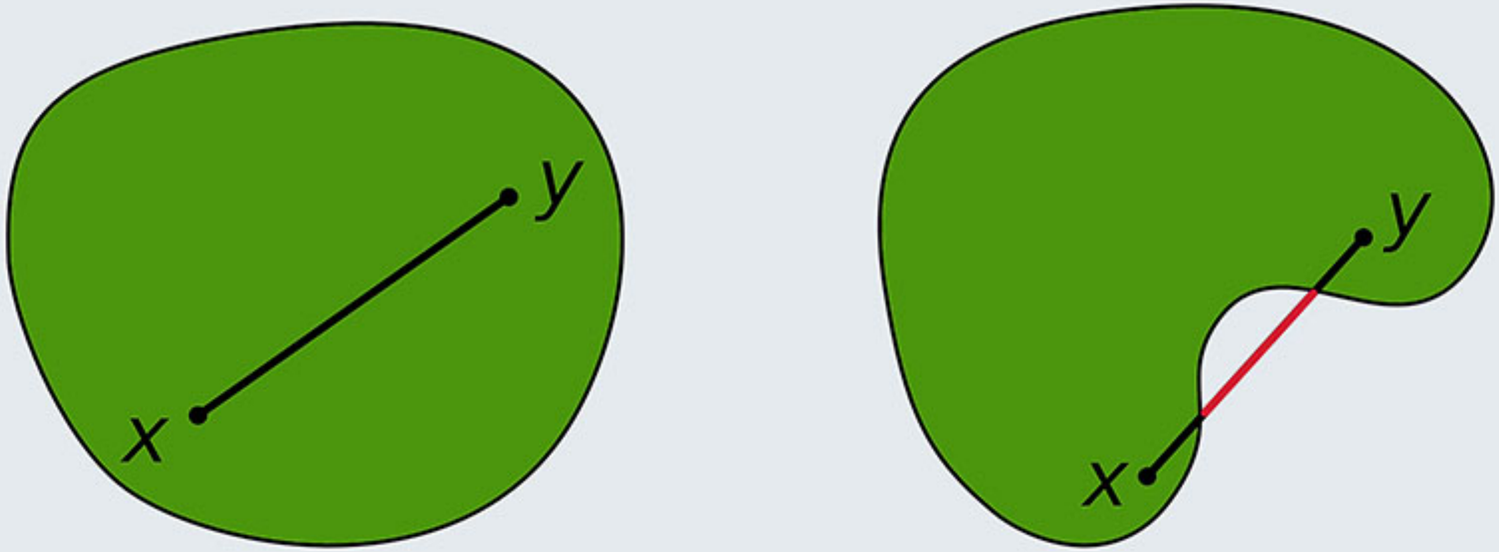
Geometrical interpretation of ILP

- the set of all points which lie in their respective feasible region of the space build a convex polytope



What is a polytope?

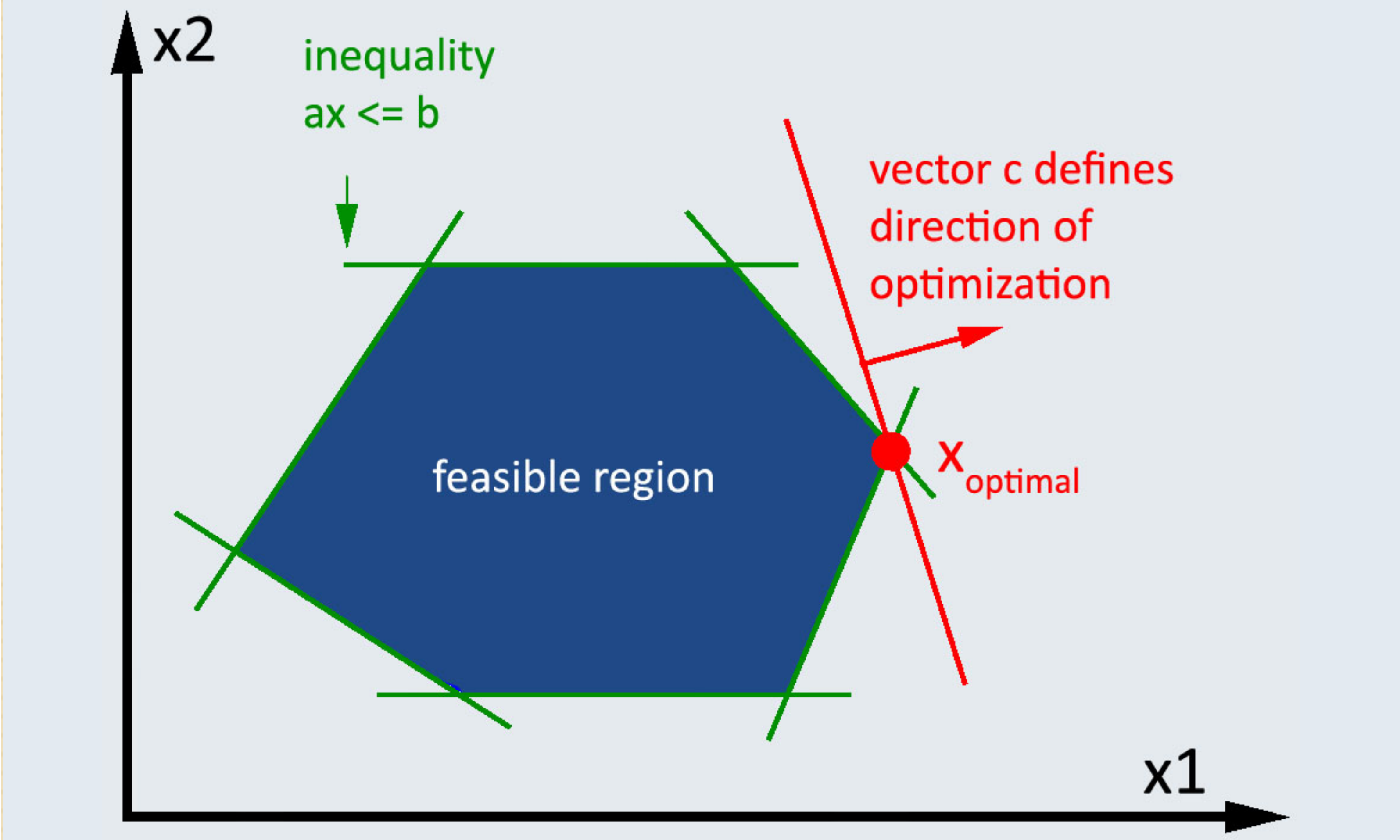
- a polytope is a closed solid geometric object with flat sides, which exists in any general number of dimensions
- 2-polytopes are called polygons
- 3-polytopes are called polyhedrons



21

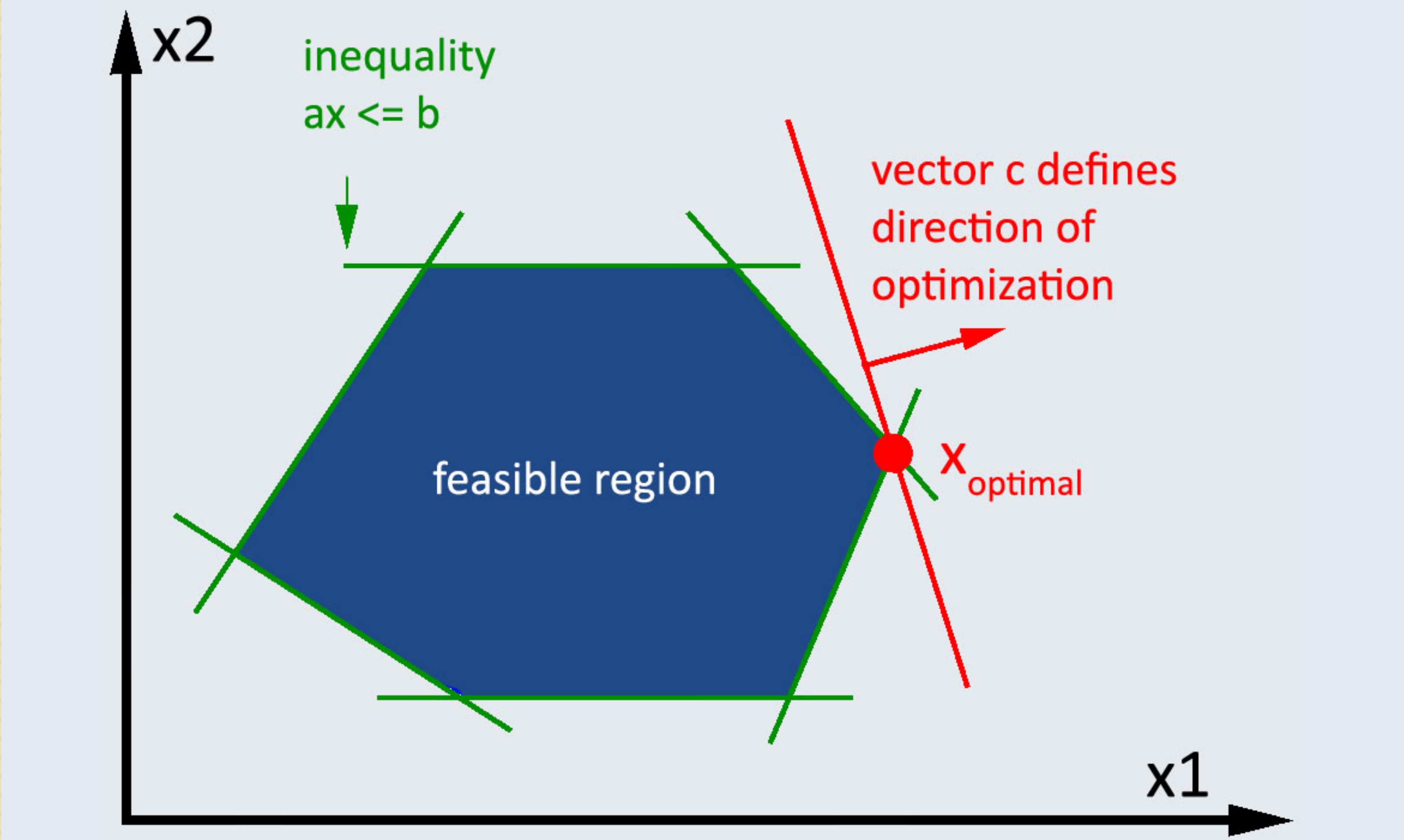
What does *convex* mean?

an object is convex if for every pair of points x and y within the object, every point on the straight line, which connects x and y, is also within the object



Geometrical interpretation of ILP

- the set of all points which lie in their respective feasible region of the space build a convex polytope
- the red vector c is the objective function c: x -> $c^T x$ that shall be maximized (standard form)
- this corresponds to finding the point in the polytope that maximizes the scalar product of vector c and the hyperplane with the condition $\{x \mid c^T x=0\}$ (the straight red line)



Geometrical interpretation of ILP

- shift this hyperplane in the direction of vector c as far as possible without leaving the polytope
- i.e. the point $x_{optimal}$ maximizes the objective function c considering the linear constraints and thus is the only optimal solution to the given problem
- in this case only one point matches, but it can also be a set of points in general

$$\hat{c}^{1:M} = \underset{c^{1:M} \in \mathcal{F}(\mathcal{P}^M)}{\operatorname{argmax}} \sum_{i=1}^{M} \operatorname{score}(S^i = c^i)$$

ILP applied to Semantic Role Labeling

$$\underset{u_{ic}\in\{0,1\}:\forall i\in[1,M],c\in\mathcal{P}}{\operatorname{argmax}} \sum_{i=1}^{M}\sum_{c\in\mathcal{P}} p_{ic}u_{ic}$$

$$\text{subject to} \quad \sum_{c\in\mathcal{P}} u_{ic} = 1 \qquad \forall i \in [1, M]$$

ILP applied to Semantic Role Labeling

the original function is reformulated as a linear function over several binary variables u_{ic} and the filter function F is represented using linear (in)equalities

$$\underset{u_{ic}\in\{0,1\}:\forall i\in[1,M],c\in\mathcal{P}}{\operatorname{argmax}} \sum_{i=1}^{M} \sum_{c\in\mathcal{P}} p_{ic} u_{ic}$$

$$\text{subject to} \quad \sum_{c\in\mathcal{P}} u_{ic} = 1 \qquad \forall i \in [1, M]$$

ILP applied to Semantic Role Labeling

- u_{ic} = [S^i = c] (indicator variable that represents whether or not the argument type c is assigned to S^i)
- p_{ic} = score(S^i = c) (the probability of argument S^i to be of type c)
- the constraint means that each argument can take only one type

ILP applied to Semantic Role Labeling

- previous approaches rely on dynamic programming to resolve non-overlapping / embedding constraints
- but they are not able to handle more expressive constraints which take, e.g., long-distance dependencies into account

Full vs. Shallow Syntactic Parsing

Full vs. Shallow Syntactic Parsing

- PropBank sections 02-21 used as training data, section 23 used for testing
- goal: understand the effective contribution of full versus shallow parsing information (i.e. using only POS tags, chunks and clauses)
- comparison of performance when using the correct (gold-standard) data versus automatic parse data
- performance is measured in terms of precision, recall and F_1 measure

Full vs. Shallow Syntactic Parsing

Full Parsing

- automatic full parse trees are derived using Charniak's parser (2001)

Shallow Parsing

- different components are used:
  - POS tagger (Even-Zohar & Roth 2001)
  - Chunker (Punyakanok & Roth 2001)
  - Clauser (Carreras et al. 2005)

The overall system performance when argument boundaries are known.

	Full Parsing			Shallow Parsing		
	Prec	Rec	F_1	Prec	Rec	F_1
Gold	91.58	91.90	91.74 ± 0.51	91.14	91.48	91.31 ± 0.51
Auto	90.71	91.14	90.93 ± 0.53	90.50	90.88	90.69 ± 0.53

Argument classification performance

- for testing, it is assumed that the argument boundaries are known
- difference between systems lies in features that are used (in the shallow system, most features can be approximated using chunks and clauses)

The performance of argument identification after pruning (based on the gold standard full parse trees).

	Full Parsing			Shallow Parsing		
	Prec	Rec	F_1	Prec	Rec	F_1
Gold	96.53	93.57	95.03 ± 0.32	93.66	91.72	92.68 ± 0.38
Auto	94.68	90.60	92.59 ± 0.39	92.31	88.36	90.29 ± 0.43

Argument identification performance

- the candidate list used here is the output of the pruning heuristic applied on the gold-standard parse trees
- difference between systems lies only in the construction of some features

The overall system performance.

	Full Parsing			Shallow Parsing		
	Prec	Rec	F_1	Prec	Rec	F_1
Gold	86.22	87.40	86.81 ± 0.59	75.34	75.28	75.31 ± 0.76
Auto	77.09	75.51	76.29 ± 0.76	75.48	67.13	71.06 ± 0.80

Pruning performance

- the main contribution of full parsing is the pruning stage
- internal tree structure significantly helps in discriminating argument candidates
- the shallow parsing system does not have enough information for the pruning heuristics, thus two word-based classifiers are trained instead (one to predict the beginning of an argument, the second to predict the end)

The impact of removing most constraints in overall system performance.

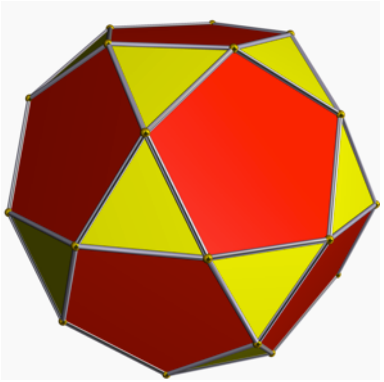
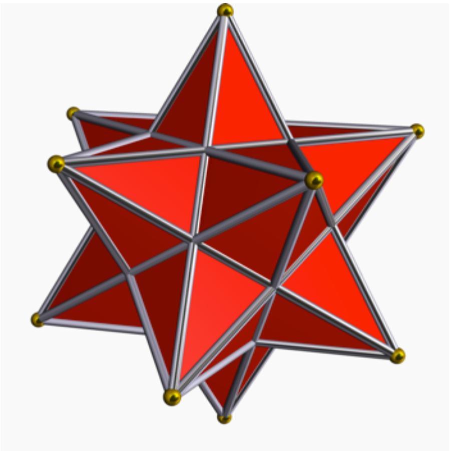
	Full Parsing			Shallow Parsing		
	Prec	Rec	F_1	Prec	Rec	F_1
Gold	85.07	87.50	86.27 ± 0.58	73.19	75.63	74.39 ± 0.75
Auto	75.88	75.81	75.84 ± 0.75	73.56	67.45	70.37 ± 0.80

ILP Inference performance

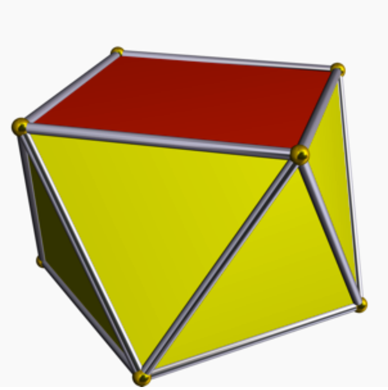
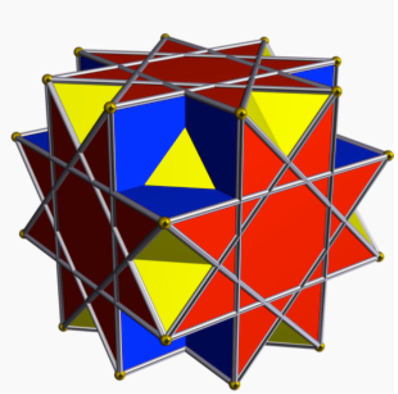
removing most linear constraints has a visible impact on performance

Conclusion

- the pruning step and the ILP-based inference procedure have the greatest impact on the overall system performance
- by means of these features a high-performant SRL system can be built
- shallow syntactic parsing yields already very good results, full syntactic parsing is most relevant in pruning and argument identification

Merci !

References

- Carlson, Andrew J.; Cumby, Chad M.; Rosen, Jeff L.; Roth, Dan (1999): *The SNoW learning architecture.* Technical Report UIUCDCS-R-99-2001, UIUC Computer Science Department.
- Carreras, Xavier; Màrquez, Llúis (2005): *Introduction to the CoNLL-2005 shared task: Semantic Role Labeling.* In: Proceedings of the Ninth Conference on Computational Natural Language Learning, p. 152-164, Ann Arbor, MI.
- Charniak, Eugene (2001): Immediate-head parsing for language models. In: Proceddings of the 39th Annual Meeting of the ACL, p. 116-123, Toulouse, France.
- Even-Zohar, Yair; Roth, Dan (2001): A sequential model for multi-class classification. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-2001), p. 10-19, Pittsburgh, PA.
- Gildea, Daniel; Jurafsky, Daniel (2002): Automatic Labeling of Semantic Roles. Computational Linguistics, 28(3):245-288.
- Punyakanok, Vasin; Roth, Dan; Yih, Wen-tau; Zimak, Dav (2004): Semantic role labeling via integer linear programming inference. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING), p. 1346-1352, Geneva, Switzerland.
- Roth, Dan (1998): Learning to resolve natural language ambiguities: A unified approach. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), p. 806-813, Madison, WI.
- Xue, Nianwen; Palmer, Martha (2004): Calibrating features for semantic role labeling. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004), p. 88-94, Barcelona, Spain.