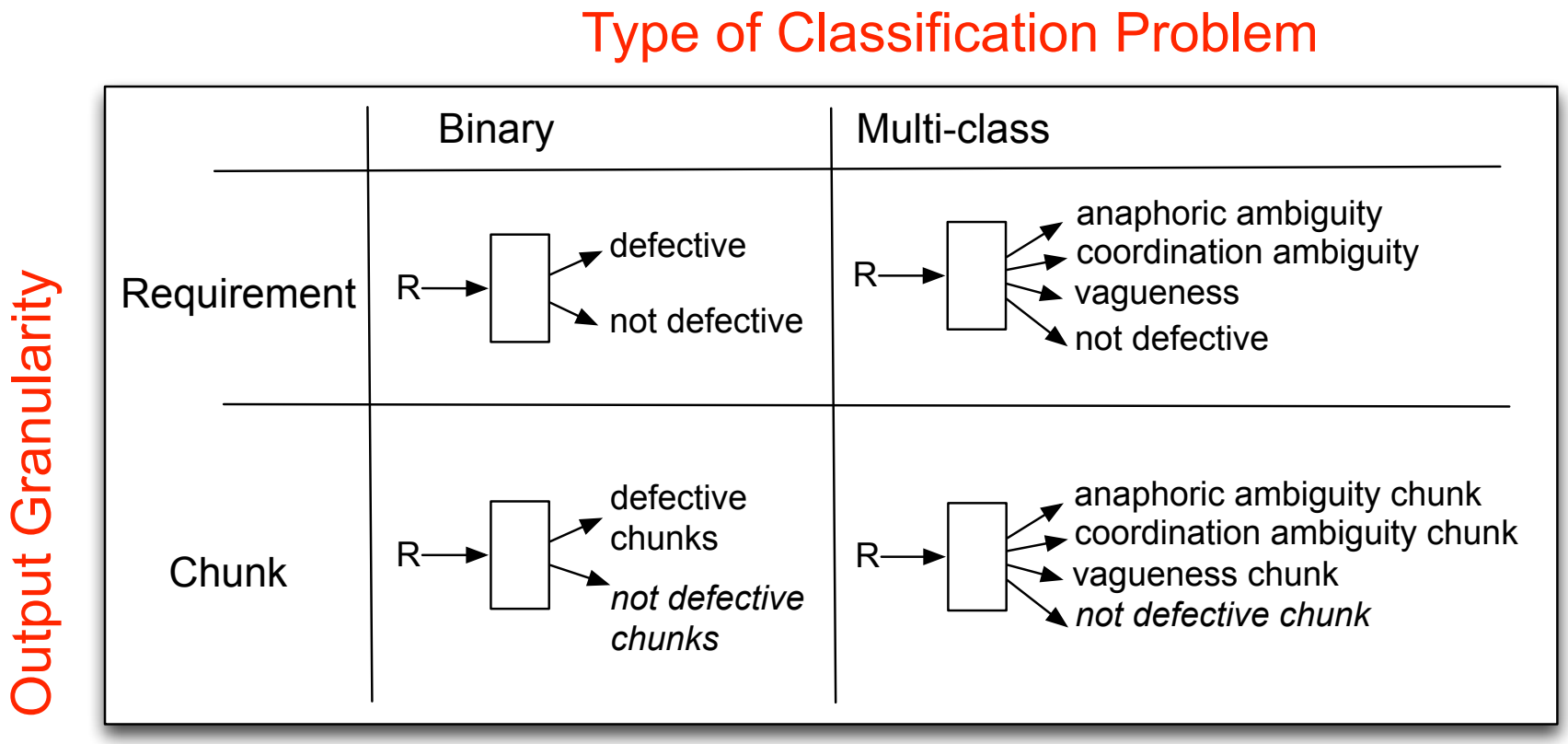
Context: Defect Detection Task

Alessio Ferrari

ISTI-CNR, Pisa, Italy
alessio.ferrari@isti.cnr.it

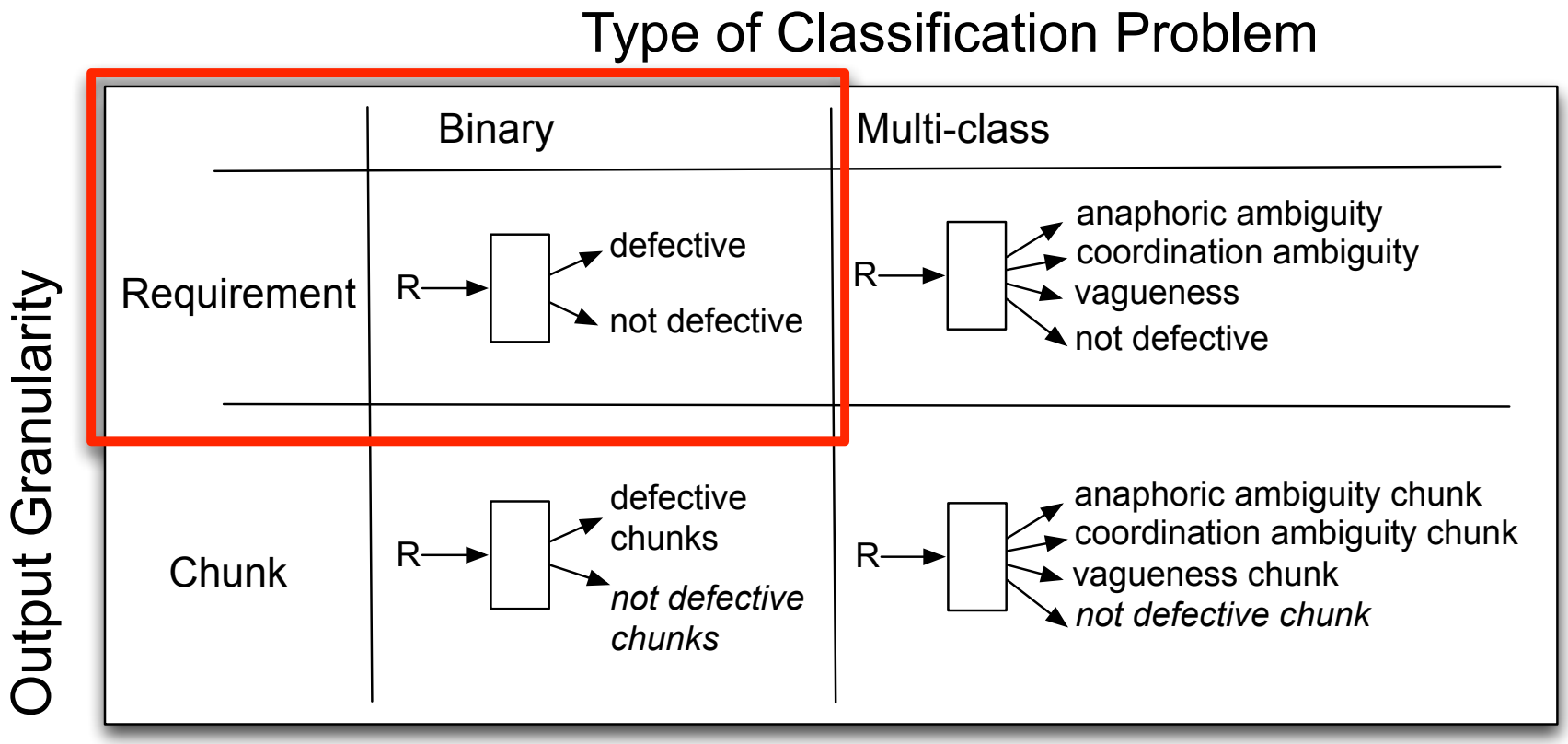
Context

Task T**:** defect detection in natural language requirements – a classification problem (many, actually)



Context

Task T**:** defect detection in natural language requirements – a classification problem (many, actually)



Recall vs Precision

- Of course recall counts more than precision ($\beta > 1$ for T)
- But how much? This cost is something that should take into account time to discard false positives, impact on the development process of false negatives, *etc.*
- Let's imagine I managed to compute $\beta = 1.7$ for T with the overview method, which focuses on time aspects

My tool t for T

- I develop my tool t for T
- I find that my t has $P = 0.6$, $R = 0.9$, $F_{1.7} = 0.8$

What can I say? **Is t GOOD or BAD?**

My tool t for T

- I develop my tool t for T
- I find that my t has $P = 0.6$, $R = 0.9$, $F_{1.7} = 0.8$

What can I say? **Is t GOOD or BAD?**

- Let's say I have a Gold Standard of 100 requirements, and 60 are defective
- If we do the math for t we have $TP = 54$, $FP = 36$, $FN = 6$, $TN = 4$

What about a tool that returns all requirements as defective?

Another imaginary tool called "All Defects"

- 100 requirements, and 60 are defective
- Imagine a tool t' that returns all requirements as defective
- I have $P = 0.6$, $R = 1$, $F_{1.7} = 0.85$

$$\rightarrow \text{My tool } t \ (F_{1.7} = 0.8) \text{ is } \textbf{BAD}!$$

- Evaluation depends on the GOLD STANDARD
- Evaluation is useless if I do not consider other BASELINES

Baseline: "All Defects"

- Equivalent to doing the task manually
- I have to check all the requirements

$$P = \frac{defective}{all} \qquad R = \frac{defective}{defective} = 1$$

Baseline: "No Defect"

- Equivalent to not doing the task at all
- I assume that requirements are correct

$$P = 0 \qquad R = 0$$

...to compare T with this baseline F-measure is not sufficient, although not doing the task is an option! (ask me later, I have hidden slides)

Other baselines are possible, e.g., HAHR, random predictor, existing tools

What do they do in NLP?

- **Shared Task:** a competition in which datasets are provided by the organisation
- Shared tasks in CoNLL (Computational Natural Language Learning, core A) from 1999
- Address fundamental NLP tasks that go from Chunking (NP, VP) to Discourse Parsing (relations)

Example: Shallow Discourse Parsing (CoNLL 2015)

- **Three sets of data**
- **Training:** the one you should use to train your system
- **Development:** to tune the system – closer to the blind test set
- **Blind test:** deploy the system on the remote machine, and we will run the system on this blind test set for the final ranking

Evaluation Measures?

- The winning tool is the one with highest F-measure on the blind test set
- For some tasks, e.g., **grammatical error correction** (CoNLL 2014), they used $F_{0.5}$, weighting precision twice as much as recall ($\beta = 0.5$)

My Humble Opinion

- The choice of β does not count that much, if you have a shared Gold Standard against which different tools can be evaluated
- As long as we do not have a shared Gold Standard for defect detection, it is useful to build up knowledge with industrial case studies, try to increase P and R as much as possible
- Choose $\beta = 1.5$, if you really need it

My Humble Opinion

Provide lessons learned instead of numbers only, since contextual factors are several:

- People learn new defects when using a tool
- The tool often performs only a part of the defect detection task
- The tool may not be qualified $\rightarrow$ manual inspection is needed
- Defects require different vetting effort
- Different defects may have different cost

Hidden Slide: Cost-based Evaluation...

What if I do not have the data to compute β?

I assume that the COST of a *fn* is *N* times the cost of a *fp*.
How much shall *N* be to make *T* preferable to the baselines?

		Tool	
		defective	not defective
Gold Standard	defective	V	$N \times V$
	not defective	V	0

$$\mathcal{C} = (fp + tp) \times V + fn \times (N \times V) = fp + tp + fn \times N$$

$fp_T = 10, tp_T = 30, fn_T = 5, tn_T = 35$, i.e., 80 reqs, 35 defective
$\mathcal{C}_T = 10 + 30 + 5 \times N = 40 + 5N$

$$\mathcal{C}_T < \mathcal{C}_{\mathcal{ALL-DEFECT}}, \mathcal{C}_{\mathcal{NO-DEFECT}}$$

$\mathcal{C}_{\mathcal{ALL-DEFECT}} = 45 + 35 + 0 \times N > \mathcal{C}_T \rightarrow N < 8$
$\mathcal{C}_{\mathcal{NO-DEFECT}} = 0 + 0 + 35 \times N > \mathcal{C}_T \rightarrow N > 1.33$

$1.33 < N < 8$ means that:

- IF the cost of a *fp* is slightly higher than the cost of *fn*
- AND IF the cost of a *fn* is less than 8 times the cost of a *fp*

$\rightarrow$ it is better to use T rather than:

doing the task manually (All Defects Baseline)

doing nothing (No Defect Baseline)