Multilingual Text Classification Made Easy

Fabrizio Sebastiani

(Joint work with Andrea Esuli and Alejandro Moreo Fernández)

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, IT
E-mail: fabrizio.sebastiani@isti.cnr.it

CICLING 2018, Hanoi, VN
March 19–23, 2018

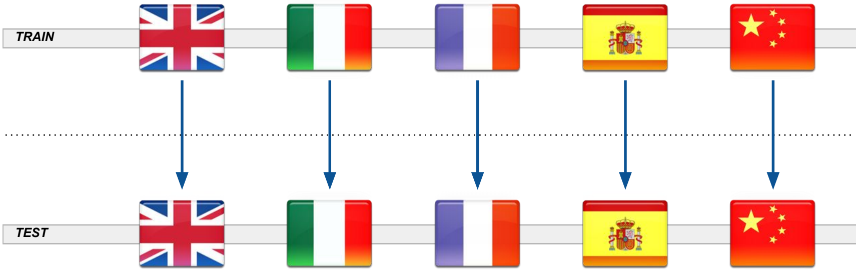
What is this talk about?

- Multilingual text classification (by topic, by sentiment, ...) ...
- ... classifier ensembles ...
- ... and transfer learning

Multilingual Text Classification

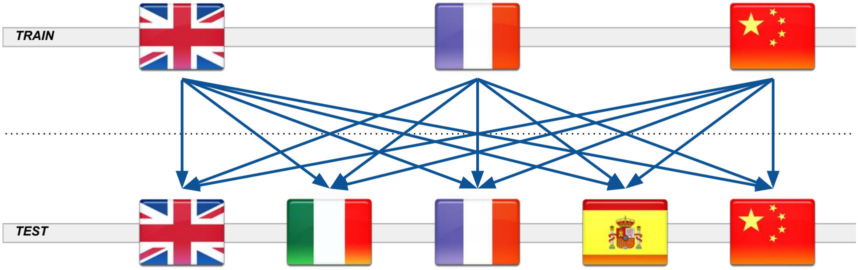
- Each document d written in one of a finite set $\mathcal{L} = \{\lambda_1, , ..., \lambda_m\}$
- Classification scheme ("codeframe") $\mathcal{C} = \{c_1, ..., c_n\}$ is the same for all languages
- Scenario common in many multinational organizations (e.g., European Union) / companies (e.g., Vodafone)
- Three "variants" of this task

1. Mono-lingual Text Classification



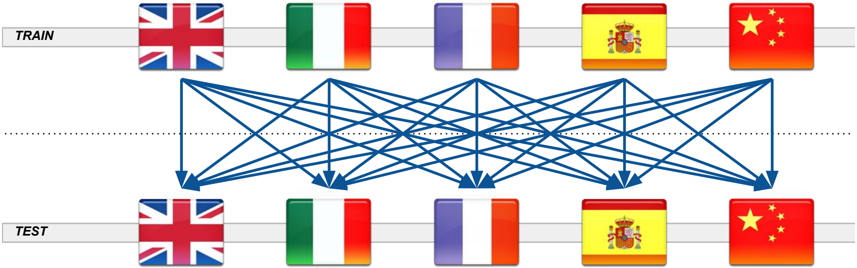
- MLC solved as m independent monolingual classification tasks

2. Cross-lingual Text Classification



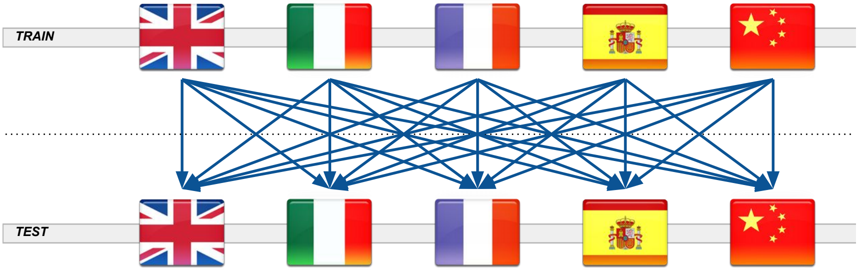
- Attempts to exploit synergies among languages
- Training examples exist only for the source languages $\mathcal{L}_s \subset \mathcal{L}$ and not for the target languages $\mathcal{L}_t \subset \mathcal{L}$
- $\Rightarrow$ Generate classifiers for languages for which you otherwise could not

3. Poly-lingual Text Classification



- Attempts to exploit synergies among languages
- Training examples exist for all languages in $\mathcal{L}$
- $\Rightarrow$ Improve on monolingual classifiers

Our problem setting



- We tackle 2 variants of polylingual multiclass classification (i.e., $n \geq 2$)
  - single-label PLC (1-of-n), which subsumes the binary case
  - multi-label PLC (any-of-n)

Classifier outputs n classification scores

Transfer Learning

- Both CLC and PLC are instances of (Heterogeneous) Transfer Learning (TL)
- Basic idea of TL: reuse info about a problem in a source domain for solving the same problem in a different target domain
- Useful to address the "training data bottleneck"
- CLC / PLC : problem = classification in $\mathcal{C}$;
  info = training examples;
  domain = language
- Useful for under-resourced languages

Transfer Learning

- PLC represents a form of massive TL : all training examples contribute to the classification of all unlabelled examples, irrespectively of language
- How can we achieve that?
- One direction is that of trying to "eliminate the differences between languages"

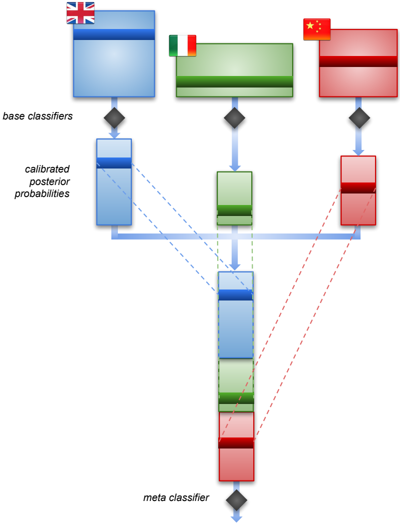
Transfer Learning

- PLC represents a form of massive TL : all training examples contribute to the classification of all unlabelled examples, irrespectively of language
- How can we achieve that?
- One direction is that of trying to “eliminate the differences between languages”

Transfer Learning

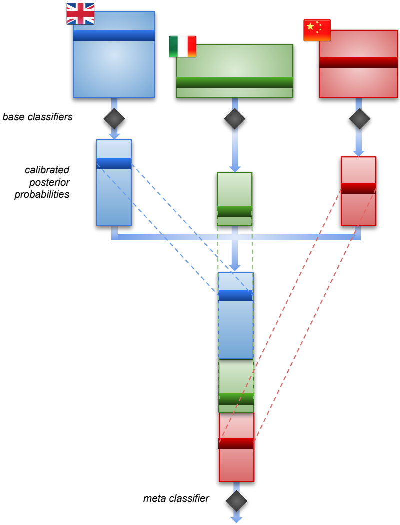
- PLC represents a form of massive TL : all training examples contribute to the classification of all unlabelled examples, irrespectively of language
- How can we achieve that?
- One direction is that of trying to "eliminate the differences between languages"
- Funnelling: a classifier ensemble method for heterogeneous TL

Funnelling: PLC made easy



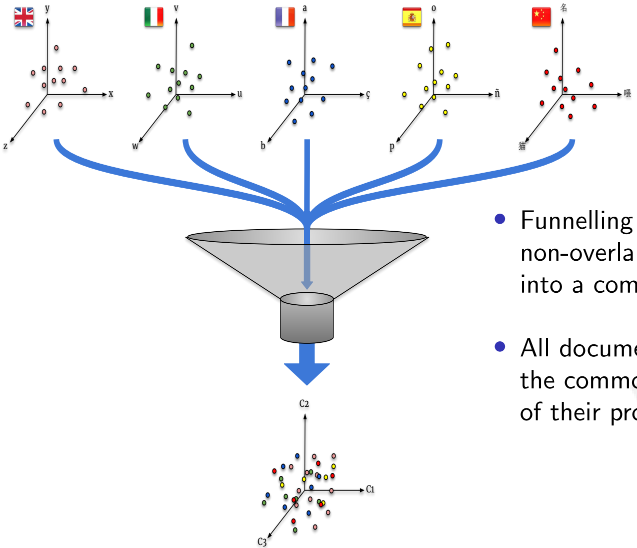
- Two-level classification architecture
  1. Set of language-dependent base classifiers
  2. Language-independent metaclassifier
- For the metaclassifier, document d represented as vector of n classification scores
- Metaclassifier outputs a vector of n classification scores

Funnelling: PLC made easy



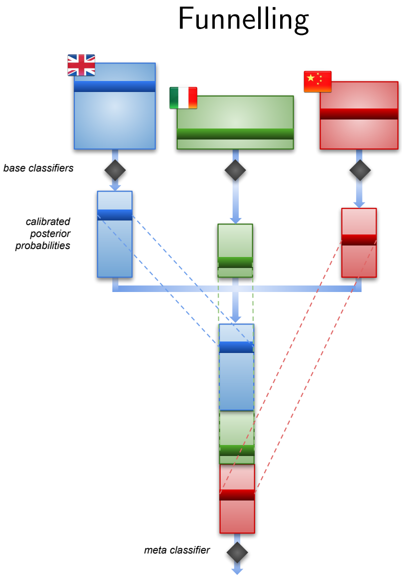
- Easy!
- Learner-independent
- Independent from representation model used in base classifiers
- No requirement that training set should be parallel or comparable
- No requirement for ML dictionaries, ML datasets, MT services

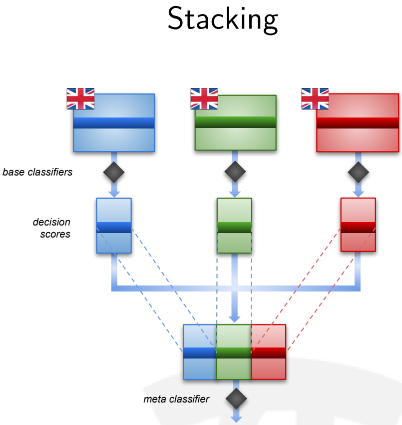
Funnelling: PLC made easy



- Funnelling maps different non-overlapping feature spaces into a common feature space
- All documents get represented in the common space irrespectively of their provenance

Funnelling vs. Stacking





Training a funnelling system

Fun(TAT) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via trained classifiers
3. Use classification scores of training examples for training metaclassifiers

Training a funnelling system

Fun(TAT) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via trained classifiers
3. Use classification scores of training examples for training metaclassifiers

- Problem: base classifiers generate higher-quality representations for training data than for test data

Training a funnelling system

Fun(TAT) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via trained classifiers
3. Use classification scores of training examples for training metaclassifiers

- Problem: base classifiers generate higher-quality representations for training data than for test data

Fun(kFCV) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via k-fold cross-validation
3. Use classification scores of training examples for training metaclassifiers

Training a funnelling system

Fun(TAT) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via trained classifiers
3. Use classification scores of training examples for training metaclassifiers

- Problem: base classifiers generate higher-quality representations for training data than for test data

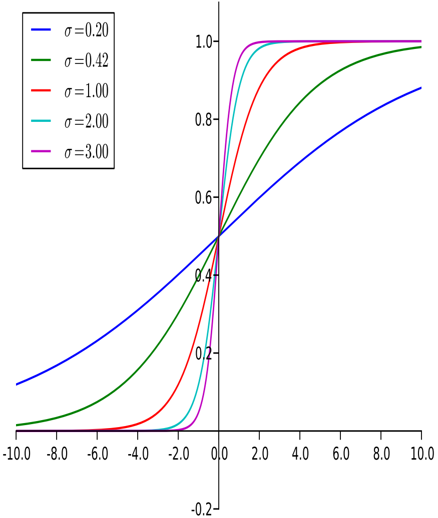
Fun(kFCV) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via *k*-fold cross-validation
3. Use classification scores of training examples for training metaclassifiers

- Problem: base classifiers generate lower-quality representations for training data than for test data

Probability calibration

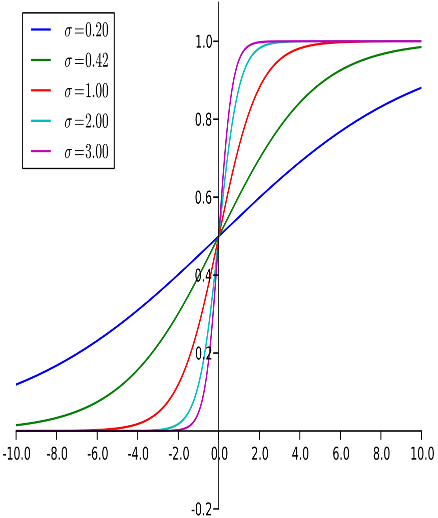
- Problem: metaclassifier receives as input vectors coming from different, incomparable sources
- Solution: make them comparable!, by converting classification scores $S(c, d)$ into well calibrated posterior probabilities $\Pr(c|d)$
- Calibration: "90% of items whose $\Pr(c|d)$ is 0.9 should belong to c"
- To be performed independently for each generated classifier



Probability calibration

- Several calibration methods available off-the-shelf (e.g., "Platt calibration")
- Needed for some learners and not for others; e.g.,

	Outputs Posterior Probs	Outputs WC Posterior Probs
SVMs	No	No
AdaBoost	No	No
Naive Bayes	Yes	No
Logistic Reg	Yes	Yes



Training a funnelling system: Fun(TAT)

Fun(TAT) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via trained classifiers
3. Map classification scores into well-calibrated posterior probabilities
4. Use posterior probabilities of training examples for training metaclassifiers

Fun(kFCV) :

1. Train base classifiers using monolingual training sets
2. Classify training examples via k-fold cross-validation
3. Map classification scores into well-calibrated posterior probabilities
4. Use posterior probabilities of training examples for training metaclassifiers

How well does funnelling work?

Datasets and learners

- Datasets:
  - RCV1/RCV2: comparable corpus, 9 languages, 10 samples $\times$ ((1000 training + 1000 test) per language), 73 classes
  - JRC-Acquis: parallel corpus, 11 languages, 10 samples $\times$ ((1155 training + 4242 test) per language), 300 classes

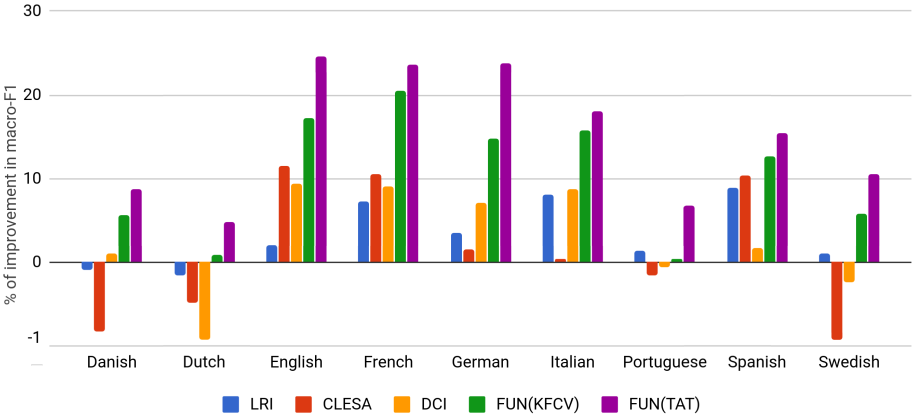
- Learners:
  - SVMs w/ linear kernel (base classifiers)
  - SVMs w/ RBF kernel (metaclassifier)

Baselines and evaluation measures

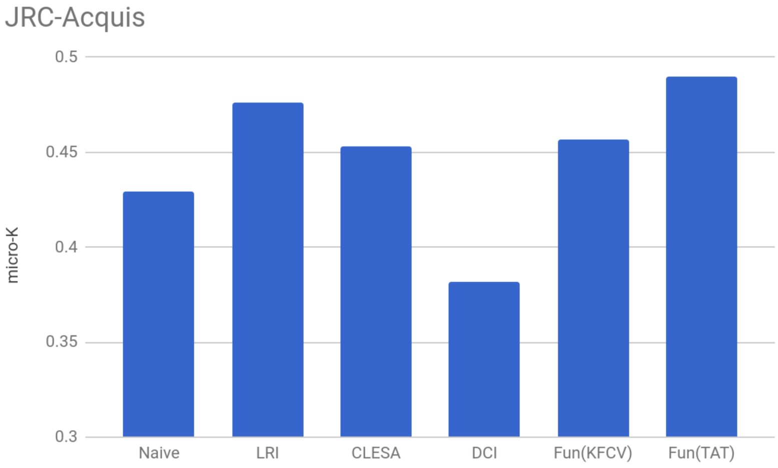
- Baselines:
  - Naïve (i.e., monolingual classification)
  - Cross-Lingual Explicit Semantic Analysis (CLESA – Song & Cimiano, CLEF 2008)
  - Distributional Correspondence Indexing (DCI – Moreo et al., JAIR 2016a)
  - Lightweight Random Indexing (LRI – Moreo et al., JAIR 2016b)
- Measures (both in micro- and macro-averaged versions):
  - F_1
  - K ($\approx$ "balanced accuracy")

Some results

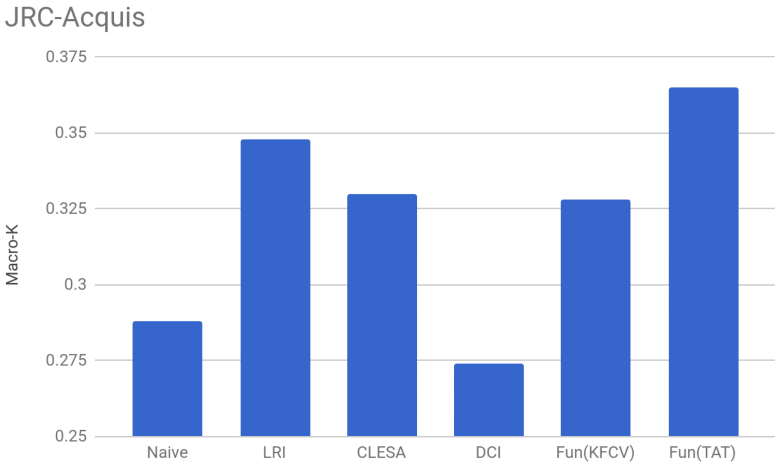
- More consistent improvements over naïve baseline



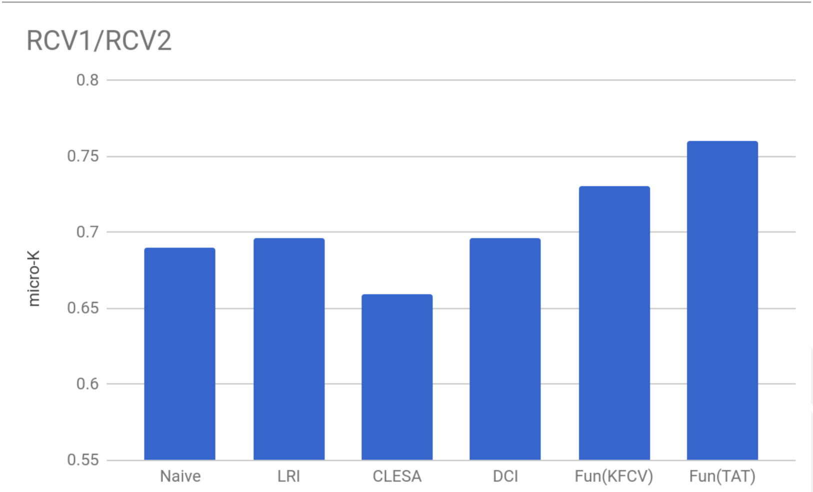
Some results: JRC-Acquis (parallel)



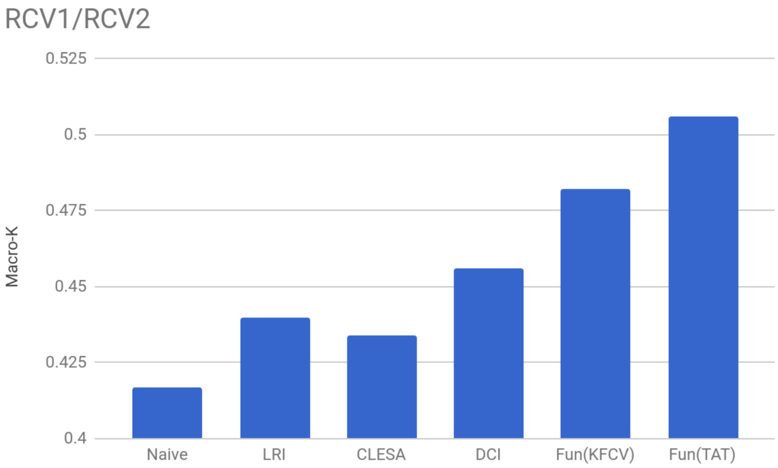
Some results: JRC-Acquis (parallel)



Some results: RCV1/RCV2 (comparable)



Some results: RCV1/RCV2 (comparable)



Overall considerations

- $\textsc{Fun}(\textsc{tat})$ significantly outperforms all other methods in 6 / 8 cases
- LRI marginally outperforms $\textsc{Fun}(\textsc{tat})$ in 2 / 8 cases
- $\textsc{Fun}(\textsc{tat})$ always outperforms $\textsc{Fun}(\textsc{kfcv})$ while being $(k+1)$ times cheaper to train
- Results for single-label PLC and multi-label PLC qualitatively similar

What does funnelling learn, exactly?

1. ~~The metaclassifier learns to combine scores from different classifiers~~

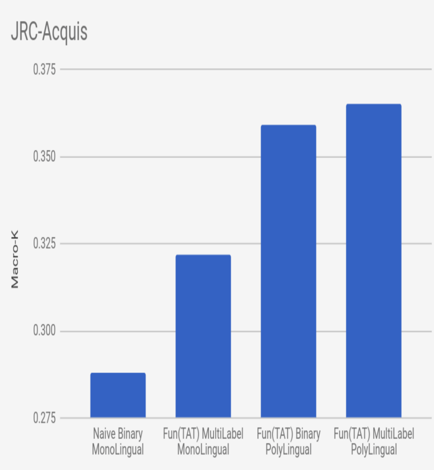
What does funnelling learn, exactly?

1. ~~The metaclassifier learns to combine scores from different classifiers~~
2. The metaclassifier learns to exploit the stochastic dependencies between classes (the multiclass factor)

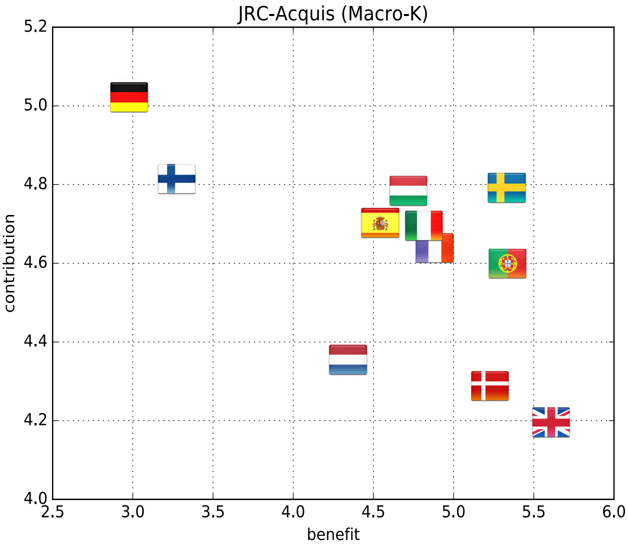
What does funnelling learn, exactly?

1. ~~The metaclassifier learns to combine scores from different classifiers~~
2. The metaclassifier learns to exploit the stochastic dependencies between classes (the multiclass factor)
3. The metaclassifier learns to classify documents in any language from training documents of any language (the multilanguage factor)

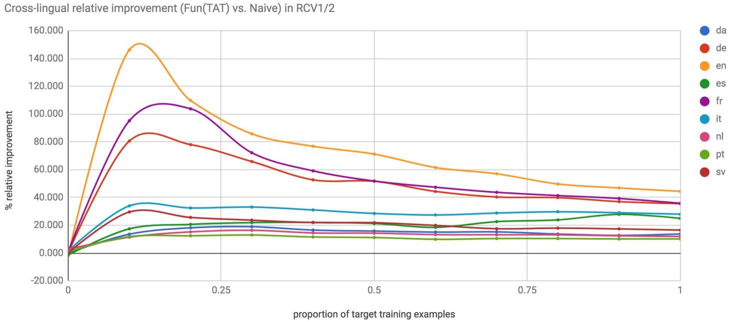
- Which factor contributes most?

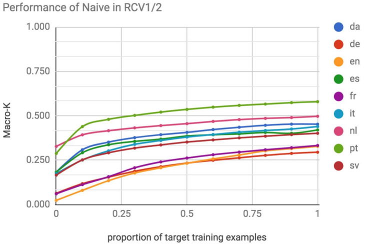


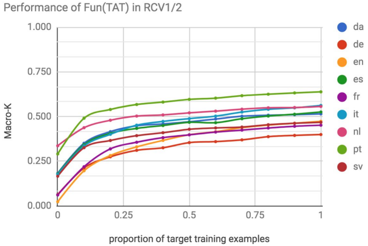
Which languages benefit / contribute most?



How does this contribution evolve?







How efficient is funnelling?

		NAÏVE	LRI	CLESA	DCI	FUN(KFCV)	FUN(TAT)
MLPLC	RCV1/RCV2	537 12	5,506 138	28,508 576	344 **3**	1,041 15	**215** 12
	JRC-Acquis	6,005 39	67,571 529	63,497 719	**4,888** **8**	13,127 54	4,987 45
SLPLC	RCV1/RCV2	285 6	3,533 61	25,187 243	130 **2**	508 8	**97** 7
	JRC-Acquis	403 2	6,048 24	9,327 32	**284** **1**	810 2	468 2

Conclusions

- PLC: an important task for many multinational organizations / companies
- Can massively benefit from transfer learning
- Approach: mapping different language-independent feature spaces into a single feature space
  - "frustratingly" easy;
  - inspired from stacking, different from it;
  - learner-independent;
  - no external resources needed (e.g., MT services, ML dictionaries, ML corpora);

Where can we go from here?

- Different codeframes ("extreme" transfer learning)
- Ordinal / hierarchical (polylingual) classification
- Other classification scenarios (e.g., "multimodal" classification)
- Supervised learning tasks different from classification (e.g., multilingual information extraction)

Questions?

Thank you!

For any question, email me at
`fabrizio.sebastiani@isti.cnr.it`

Multi-label PLC results

		Naïve	LRI	CLESA	DCI	Fun(kfcv)	Fun(tat)	UpperBound
F_1^μ	RCV1/RCV2	.776	.771	.714	.770	.801†	**.802**	–
	JRC-Acquis	.559	**.594**	.557	.510	.581	.587	.707
F_1^M	RCV1/RCV2	.467	.490	.471	.485	.512	**.534**	–
	JRC-Acquis	.340	**.411**	.379	.317	.356	.399	.599
K^μ	RCV1/RCV2	.690	.696	.659	.696	.731	**.760**	–
	JRC-Acquis	.429	.476	.453	.382	.457	**.490**	.632
K^M	RCV1/RCV2	.417	.440	.434	.456	.482	**.506**	–
	JRC-Acquis	.288	.348	.330	.274	.328	**.365**	.547

Which factor contributes most?

		Naïve Binary MonoLin	Fun(tat) MultiLab MonoLin	Fun(tat) Binary PolyLin	Fun(tat) MultiLab PolyLin
F_1^μ	RCV1/RCV2	.776	.800††	.801††	**.802**
	JRC-Acquis	.559	.573	**.589**	.587††
F_1^M	RCV1/RCV2	.467	.527	.532†	**.534**
	JRC-Acquis	.340	.366	.395††	**.399**
K^μ	RCV1/RCV2	.690	.748	.757	**.760**
	JRC-Acquis	.429	.447	.487††	**.490**
K^M	RCV1/RCV2	.417	.492	.505†	**.506**
	JRC-Acquis	.288	.322	.359	**.365**

Single-label PLC results

		Naïve	LRI	CLESA	DCI	Fun(kfcv)	Fun(tat)	UpperBound
F_1^μ	RCV1/RCV2	.759	.766	.706	.736	**.792**	.781	–
	JRC-Acquis	.202	**.353**	.331	.262	.318	.340†	.593
F_1^M	RCV1/RCV2	.538	.558	.543	.543	.584	**.596**	–
	JRC-Acquis	.362	**.407**	.400	.374	.382	.389	.570
K^μ	RCV1/RCV2	.649	.670	.636	.646	.715	**.757**	–
	JRC-Acquis	.115	.222	.215	.163	.205	**.253**	.463
K^M	RCV1/RCV2	.503	.522	.521	.527	.559	**.594**	–
	JRC-Acquis	.358	.400	.396	.380	.389	**.407**	.570

Efficiency results

		Naïve	LRI	CLESA	DCI	Fun(kfcv)	Fun(tat)
MLPLC	RCV1/RCV2	537	5,506	28,508	344	1,041	**215**
		12	138	576	**3**	15	12
	JRC-Acquis	6,005	67,571	63,497	**4,888**	13,127	4,987
		39	529	719	**8**	54	45
SLPLC	RCV1/RCV2	285	3,533	25,187	130	508	**97**
		6	61	243	**2**	8	7
	JRC-Acquis	403	6,048	9,327	**284**	810	468
		2	24	32	**1**	2	2

Table: Computation times (in seconds); 1st rows indicate training times while 2nd rows report testing times.