Differential Evaluation: a Qualitative Analysis of Natural Language Processing System Behavior Based Upon Data Resistance to Processing

Lucie Gianola, Hicham El Boukkouri, Cyril Grouin,
Thomas Lavergne, Patrick Paroubek, Pierre Zweigenbaum
Université Paris-Saclay, CNRS, LISN, 91405, Orsay, France
firstname.lastname@lisn.fr

Abstract

Most of the time, when dealing with a particular Natural Language Processing task, systems are compared on the basis of global statistics such as recall, precision, F1-score, etc. While such scores provide a general idea of the behavior of these systems, they ignore a key piece of information that can be useful for assessing progress and discerning remaining challenges: the relative difficulty of test instances. To address this shortcoming, we introduce the notion of *differential evaluation* which effectively defines a pragmatic partition of instances into gradually more difficult bins by leveraging the predictions made by a set of systems. Comparing systems along these difficulty bins enables us to produce a finer-grained analysis of their relative merits, which we illustrate on two use-cases: a comparison of systems participating in a multi-label text classification task (CLEF eHealth 2018 ICD-10 coding), and a comparison of neural models trained for biomedical entity detection (BioCreative V chemical-disease relations dataset).

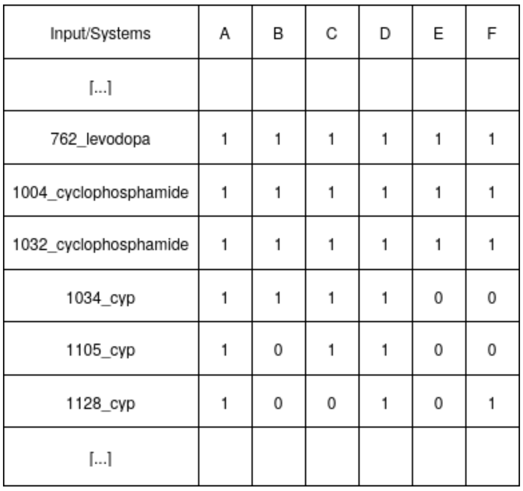
1 Introduction

The analysis of NLP system results has mainly focused on evaluation scores meant to rank systems and feed leaderboards. In tasks such as information extraction, text classification, etc., evaluation generally relies on the comparison of a hypothesis (typically a system output) with a gold standard, generally produced through manual annotation. Since the MUC-6 conference (Grishman and Sundheim, 1996), the metrics used were created for information retrieval (Cleverdon, 1960): recall (true positive rate), precision (positive predictive value) and their harmonic (possibly weighted) mean, the F1-score. Evaluation scripts are widely available nowadays, for instance those of the CoNLL shared tasks (Tjong Kim Sang and De Meulder, 2003). These scripts rely on an annotation scheme based on the BIO prefix used to specify whether a token is at the beginning, inside or outside of an annotation span, making it a *de facto* standard for NER evaluation (Nadeau and Sekine, 2007). Many other NLP tasks have developed or used their own metrics, such as accuracy for classification, BLEU (Papineni et al., 2002) for machine translation, ROUGE for machine translation and text summarization (Lin, 2004), word error rate for automatic speech recognition, etc. While evaluation is the key step in shared tasks, developers also need to evaluate the performance of their systems for feature selection or architecture design choices, especially when several systems are combined (Jiang et al., 2016).

However, scores only are insufficient to capture the behavior of systems and to provide a finer-grained analysis of their pros and cons. Indeed, though widely used, scores are not free of imperfections, as demonstrated by Peyrard et al. (2021) who discuss the use of the average to aggregate evaluation scores. They show that very different system behaviors can yield similar scores when using the average and suggest an alternative aggregation mechanism. Some researchers also call for going beyond performance scores: Ethayarajh and Jurafsky (2020) suggest that performance-based evaluation (as promoted by leaderboards) overlooks aspects such as utility, prediction cost, and robustness of models. They recommend considering the point of view of the user of models rather than just performance scores to estimate their relevance.

Trying to provide a finer understanding of the issues raised by the input text and of the limitations of the evaluated systems, we propose a new qualitative analysis method that takes into account the observed relative difficulty of predicting gold labels for each input. This difficulty is assessed pragmatically based upon the number of systems that predict a gold label (a true positive) for a given input. As a qualitative method, its aim is not to compute an evaluation measure nor to rank systems, but

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1–10
July 5–10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.26615/978-954-452-056-4_001

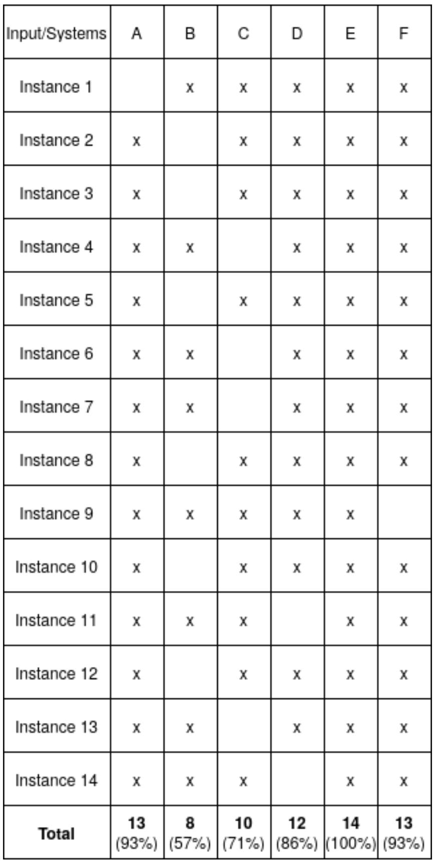


Input/Systems	A	B	C	D	E	F
[...]						
762_levodopa	1	1	1	1	1	1
1004_cyclophosphamide	1	1	1	1	1	1
1032_cyclophosphamide	1	1	1	1	1	1
1034_cyp	1	1	1	1	0	0
1105_cyp	1	0	1	1	0	0
1128_cyp	1	0	0	1	0	1
[...]						

Figure 1: Example input file for a set of six systems. 1 means the system yielded a true positive for the instance, and 0 means it did not (the instance was 'missed').

instead to obtain an overview of *how* different systems achieve the task, and thus understand where their strengths and weaknesses are.

After explaining how the method works globally (Section 2), we illustrate it with data from two shared tasks from the biomedical domain, one for multi-label classification and another for named entity recognition (Section 3), then discuss a few points and directions for future investigation (Section 4).



Input/Systems	A	B	C	D	E	F
Instance 1		x	x	x	x	x
Instance 2	x		x	x	x	x
Instance 3	x		x	x	x	x
Instance 4	x	x		x	x	x
Instance 5	x		x	x	x	x
Instance 6	x	x		x	x	x
Instance 7	x	x		x	x	x
Instance 8	x		x	x	x	x
Instance 9	x	x	x	x	x	
Instance 10	x		x	x	x	x
Instance 11	x	x	x		x	x
Instance 12	x		x	x	x	x
Instance 13	x	x		x	x	x
Instance 14	x	x	x		x	x
Total	**13** (93%)	**8** (57%)	**10** (71%)	**12** (86%)	**14** (100%)	**13** (93%)

Figure 2: Composition of bin-5 in the comparison of six systems. Each instance (row) is missed by exactly one system. Note that each system (column) may miss multiple instances in this bin.

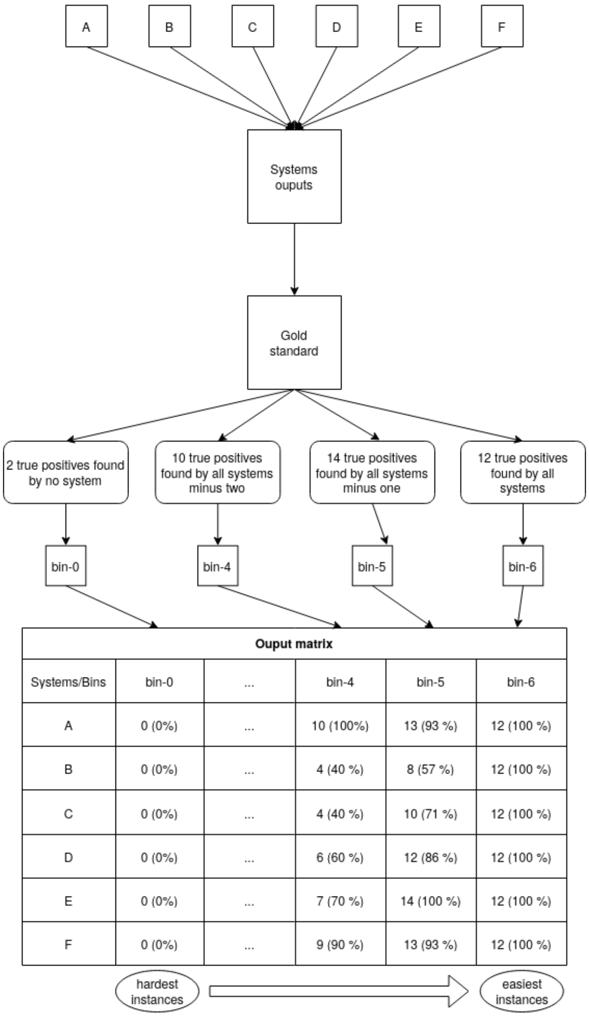
2 Differential evaluation: highlighting the 'difficulty' of examples

Our qualitative analysis method, which we call *differential evaluation*[1], globally considers the various sets of correct instances ('true positives', or 'gold instances') that were discovered by a set of systems. Since the aim of the method is not to produce a ranking, the considered systems can be different systems performing the same task, as in a shared task for example, or different versions of the same system also performing a given task, as in a development context.

As input, the algorithm takes a matrix of instances and systems, as shown in Figure 1. For each instance, it then computes how many systems discovered it correctly (i.e., in Figure 1, '762_levodopa' has been discovered by 6 systems, '1034_cyp' has been discovered by 4 systems, etc.) This enables it to compute then how many instances have been detected by all systems, by all systems but one, by all systems but two, etc., and by no system at all. This yields a grouping of instances into *bins* depending on the number of systems that discovered them. There are as many bins as there are systems plus one for the set of instances that were discovered by none of the systems. Bin-1 is the set of instances detected by exactly one system, bin-2 the set of instances detected by exactly two systems, etc.; and bin-0, the set of instances that no system was able to detect (see Section 3 for illustrated examples). Figure 2 shows the composition of bin-5 in a case where six systems are compared, and displays the percentage coverage of the bin for each system. Figure 3 shows a schema of the global scenario of the method.

Instances in bin-N (where N is the number of considered systems), which holds the set of entities discovered by all systems, can be considered as the *easiest* to predict, while instances in bin-0, which holds the set of entities that no system was able to detect, can be seen as the *most difficult*. More-

[1] https://github.com/PierreZweigenbaum/differential-evaluation


A
B
C
D
E
F
Systems ouputs
Gold standard
2 true positives found by no system
10 true positives found by all systems minus two
14 true positives found by all systems minus one
12 true positives found by all systems
bin-0
bin-4
bin-5
bin-6
hardest instances
easiest instances

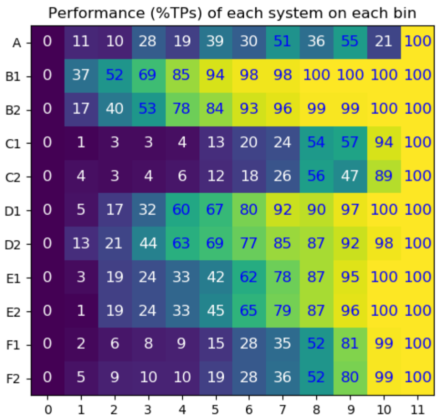
Ouput matrix

Systems/Bins	bin-0	...	bin-4	bin-5	bin-6
A	0 (0%)	...	10 (100%)	13 (93 %)	12 (100 %)
B	0 (0%)	...	4 (40 %)	8 (57 %)	12 (100 %)
C	0 (0%)	...	4 (40 %)	10 (71 %)	12 (100 %)
D	0 (0%)	...	6 (60 %)	12 (86 %)	12 (100 %)
E	0 (0%)	...	7 (70 %)	14 (100 %)	12 (100 %)
F	0 (0%)	...	9 (90 %)	13 (93 %)	12 (100 %)

Figure 3: Differential evaluation scenario.
True positives (TPs) are displayed with absolute and relative values (percentage of the number of instances in the bin) in the output matrix, as in Table 1 and Figure 4 respectively. System contributions to a bin can have a null intersection: i.e. here, in bin-4, Systems B and C may be yielding TPs for totally different sets of instances. Bins 1, 2 and 3 omitted for conciseness.

over, bin-1, which holds instances discovered by a single system, can be seen as the bin holding the singular contribution of each system. As such, bin-1 is particularly interesting when considering system combination architectures or ROVER-like performance measures (Fiscus, 1997).

Figure 4 presents one of the outputs of the method, a heatmap of percentages of system TPs relative to the total number of instances in each bin, in this case for the CLEF eHealth 2018 ICD-10 coding task for Italian (we analyse this example in detail in Section 3.1.1). The first column on the left is bin-0, holding only 0 values as we have said that bin-0 is the bin of instances missed by all systems (as shown by Table 1, here 305 instances were missed by all systems). The second column from the left holds bin-1, and so on. Another output of the method is the table of absolute values corresponding to the percentages heatmap, such as Table 1. It would then be interesting to investigate whether a pattern emerges concerning the linguistic nature of instances, which would help to chart the difficulty of the task, and complete the qualitative aspect of the analysis.



	0	1	2	3	4	5	6	7	8	9	10	11
A	0	11	10	28	19	39	30	51	36	55	21	100
B1	0	37	52	69	85	94	98	98	100	100	100	100
B2	0	17	40	53	78	84	93	96	99	99	100	100
C1	0	1	3	3	4	13	20	24	54	57	94	100
C2	0	4	3	4	6	12	18	26	56	47	89	100
D1	0	5	17	32	60	67	80	92	90	97	100	100
D2	0	13	21	44	63	69	77	85	87	92	98	100
E1	0	3	19	24	33	42	62	78	87	95	100	100
E2	0	1	19	24	33	45	65	79	87	96	100	100
F1	0	2	6	8	9	15	28	35	52	81	99	100
F2	0	5	9	10	10	19	28	36	52	80	99	100

Figure 4: Percentage of labels (true positives) correctly found by each system in each bin for Italian in the CLEF eHealth 2018 ICD-10 coding task. Systems on x-axis and bins on y-axis.

3 Experiments

In this section, we present insights that can be drawn from the use of differential evaluation on data related to two shared tasks addressing respectively multi-label text classification and named entity recognition, both in the biomedical domain. Note that our algorithm processes the systems in the order in which they are presented and that it is not intended to create a new ranking of the systems, but rather to provide more fine-grained information to analyze how a given system has performed or achieved its ranking.

3.1 CLEF eHealth 2018 ICD-10 coding

We show as an example the output obtained in the comparison of systems in a multi-label text classification task in Italian and Hungarian (Névéol et al., 2018). In the gold standard, each input text is associated to one or more true labels, i.e., codes in the International Classification of Diseases (ICD-10). A true positive system prediction is an association between a given input text and one of the true labels for this text in the gold standard. In this dataset,

Systems	bin-0	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	bin-10	bin-11	Total TPs per system
A	0	21	33	93	104	271	311	652	*645*	765	*829*	3800	*7524*
B1	0	**69**	**163**	**224**	**472**	**648**	**1005**	**1245**	**1774**	**1390**	**3890**	3800	**14680**
B2	0	31	126	172	434	575	959	1211	1760	1373	3886	3800	14327
C1	0	*2*	*8*	*11*	*24*	89	208	*306*	958	813	3658	3800	9877
C2	0	7	11	14	31	*83*	*189*	327	1005	*660*	3445	3800	9572
D1	0	9	55	105	331	463	823	1168	1608	1344	3884	3800	13590
D2	0	24	67	143	351	474	795	1073	1543	1284	3827	3800	13381
E1	0	6	60	77	183	289	639	982	1549	1327	3886	3800	12798
E2	0	*2*	60	78	184	312	665	1003	1557	1337	3886	3800	12884
F1	0	4	20	27	49	105	291	444	919	1125	3854	3800	10638
F2	0	10	29	34	57	131	289	458	930	1110	3855	3800	10703
Total per bin	305	185	316	326	555	688	1029	1267	1781	1392	3890	3800	15534

Table 1: Number of labels (true positives) correctly found by each system in each bin for Italian: absolute values. Bin n contains the labels found by exactly n systems. Best performance in green, worst performance in red.

the evaluation method therefore compares label attribution rather than entities.

3.1.1 Italian

Eleven systems were examined for Italian, and 15,534 labels were to be discovered. Some of the teams that participated in the shared task submitted two runs for variants of their base system, hence names such as B1 and B2 when two systems are submitted by the same team in Figure 4 and other tables or figures. As shown in Table 1, bin-0 holds 305 labels found by none of the systems. Bin-1 holds 185 labels found by exactly one system, among which System A discovered 21 labels, System B1 discovered 69 labels, and so on. Bin-11 holds 3,800 labels found by all eleven systems. Figure 4 and Table 1 show bin repartition with percentages and absolute values. In Table 1, column “Total TPs per system” presents the total number of labels found per system, and row “Total per bin” contains the total number of labels to be found. We use color codes to highlight the best/worst system for each bin.

Performances are pretty steady, with System B1 outperforming all the others in every bin. The worst results are shared by Systems C1 and C2, and System A that performs badly for bins-8 and 10, which are among the “easiest” bins. As seen in Table 1, although System E2 scores the worst for bin-1 with only two labels discovered, it manages to keep up with the performances of the other systems in the other bins, and its global performance (12,884 total TPs discovered) is pretty average. On the other hand, Systems C1 and C2, which are the worst systems across all bins, are not so bad globally with 9,877 and 9,572 total TPs. In fact, System A achieves a very low performance on two of the “easiest” bins, and thus yields less than half of the total labels, despite a not so bad performance on bin-1. Figure 4 shows that systems can be divided into groups of better and worse performances (B1, B2, D1, D2, E1, E2 vs. A, C1, C2, F1, F2). We can also see that System B1 reaches a perfect score over all easier bins up to bin-8, which hints at its being robust on easy instances.

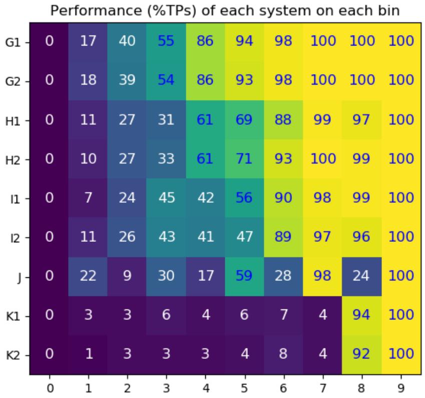


Figure 5: Percentage of labels (true positives) correctly found by each system in each bin for Hungarian. Systems on x-axis and bins on y-axis.

3.1.2 Hungarian

Figure 5 and Table 2 show the proportion and number of detected labels per system within each bin for the Hungarian language[2].

As highlighted by colors in Table 2, we can see that globally, Systems G1 and G2 perform the best, and Systems K1 and K2 perform the worst.

Just above K1 and K2 in terms of Total TPs per system (Table 2), System J is the worst at detecting labels from bin-8 (see also Figure 5), which can

[2] The data are not the same as that for Italian, hence the different total values.

Systems	bin-0	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	Total TPs per system
G1	0	104	**555**	**655**	**2855**	**4760**	**11246**	37654	**9034**	26324	**93187**
G2	0	116	542	642	2828	4700	11239	**37659**	9028	26324	93078
H1	0	72	381	375	2001	3471	10080	37367	8749	26324	88820
H2	0	62	380	394	2019	3606	10620	37525	8985	26324	89915
I1	0	45	333	538	1375	2877	10293	36748	8980	26324	87513
I2	0	67	366	519	1356	2400	10208	36575	8695	26324	86510
J	0	**136**	126	364	557	2986	3331	37024	*2134*	26324	72982
K1	0	19	46	73	140	285	*832*	1693	8460	26324	37872
K2	0	*7*	*45*	*40*	*89*	*215*	947	*1508*	8303	26324	*37478*
Total per bin	1442	628	1387	1200	3305	5060	11466	37679	9046	26324	97537

Table 2: Number of labels (true positives) correctly found by each system in each bin for Hungarian: absolute values. Bin n contains the labels found by exactly n systems. In this analysis, the systems are ordered in decreasing order of F1-score, determined prior to the present analysis.

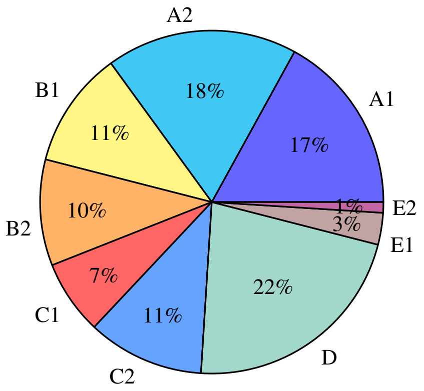


Figure 6: Proportion of labels discovered by exactly one system, per system for Hungarian.

be considered "easy" labels, with a very low proportion of 24% when all other systems are above 90%. In contrast however, it detects the largest number of labels in bin-1 (see also Figure 6). This is the only case where System G1 is significantly outperformed. System J is therefore good at detecting some "difficult" labels. This is a strong indicator that this system is likely to use a method that is quite different from the other systems and might bring complementary expertise on some inputs, which deserves further investigation.

Another perspective comes from looking at the overall performance for labels from bin-1, which, contrary to the example of Italian where most of bin-1 is yielded by four systems among eleven, is distributed in a more balanced way among systems. This means that labels from bin-1 are not yielded by one unique system that would be outperforming all the others, but that every system makes an important contribution to this bin (Figure 6).

3.2 BioCreative V CDR entities

The BioCreative V chemical-disease relation (CDR) task is originally a relation extraction task (Wei et al., 2016). Its data can also be used to train and evaluate entity-detection systems for chemical and disease entities, which is what we examine here. The dataset is made of 1,500 PubMed abstracts of scientific papers, divided equally into training, development and test. In the gold standard, each input token is associated to one true label and named entities are encoded according to the BIO (begin, inside, outside) scheme. In the present work we deal with tokens rather than entities, so that we can apply the presented method directly. We consider that 'O' labels are negatives and that all other labels are positives. A true positive system prediction is an association between an input token and a non-'O' label that is the gold-standard label for this token.

We are comparing entity detection systems that rely on word embeddings based upon CharacterBert (El Boukkouri et al., 2020) or fastText (Bojanowski et al., 2017), pre-trained on different corpora, either *as-is* or concatenated with knowledge embeddings learned using node2vec (Grover and Leskovec, 2016) on two biomedical vocabularies (the Medical Suject Headings (MeSH), and SNOMED CT). Moreover, we also consider a variant of CharacterBert where the node2vec embeddings are injected within the model architecture. The fastText embeddings are either randomly initialized, which we note "fastTextRandom"; pre-trained on a newswire corpus (Gigaword (Graff et al., 2007)), which we note "fastTextGigaword"; or on medical corpora (PubMed Central[3] and MIMIC-III (Johnson et al., 2016)), which

[3] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

Model	bin-0	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	bin-10	bin-11	bin-12	Tot. TPs /system
Enh.CharBertFromGenN2V	0	**12**	65	72	**155**	**148**	156	**176**	223	**294**	465	852	3894	6512
CharBertFromGen	0	9	**70**	**75**	147	147	**158**	174	**228**	287	**477**	868	3894	**6534**
CharBertGenN2V	0	1	10	41	107	112	139	168	199	282	466	868	3894	6287
CharBertGen	0	3	7	41	103	113	131	163	205	285	463	853	3894	6261
fastTextGigawordN2V	0	6	7	*7*	28	61	77	110	164	244	446	**869**	3894	5913
fastTextGigaword	0	*0*	*3*	*7*	*19*	60	78	111	*106*	196	*343*	*812*	3894	*5629*
fastTextMimicN2V	0	*0*	9	14	29	*43*	59	91	165	235	450	862	3894	5851
fastTextMimic	0	2	10	9	20	53	*56*	*88*	128	*190*	413	830	3894	5693
fastTextPubMedN2V	0	4	12	21	47	51	87	113	190	254	453	830	3894	5956
fastTextPubMed	0	3	10	29	39	83	101	116	182	247	449	862	3894	6015
fastTextRandomN2V	0	0	5	11	28	39	39	77	106	161	322	792	3894	5474
fastTextRandom	0	1	2	9	18	30	41	62	56	106	143	338	3894	4700
Total TPs per bin	178	41	105	112	185	188	187	207	244	309	489	876	3894	7015

Table 3: Absolute values for chemical NER. Best performance in green, worst performance in red, orange when the random initialization is above one of the other initializations.

Models	bin-0	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	bin-10	bin-11	bin-12	Tot. TPs /system
Enh.CharBertFromGenN2V	0	16	70	74	124	115	159	**181**	**256**	**296**	389	800	3617	6097
CharBertFromGen	0	**44**	**89**	**92**	**142**	**123**	**164**	179	247	289	389	791	3617	**6166**
CharBertGenN2V	0	14	29	66	106	110	137	166	238	278	378	795	3617	5934
CharBertGen	0	24	32	57	110	107	137	162	234	287	387	802	3617	5956
fastTextGigawordN2V	0	*3*	22	36	59	70	112	141	224	288	403	803	3617	5778
fastTextGigawordN2V	0	5	*7*	*17*	*25*	*50*	*72*	*91*	*126*	205	*311*	*730*	3617	*5256*
fastTextMimicN2V	0	6	12	25	39	54	103	144	207	257	359	791	3617	5614
fastTextMimic	0	13	12	29	33	51	85	94	145	*200*	325	746	3617	5350
fastTextPubMedN2V	0	6	15	32	64	65	141	162	236	292	**408**	**814**	3617	5852
fastTextPubMed	0	5	12	29	50	53	103	118	182	204	332	764	3617	5469
fastTextRandomN2V	0	10	27	41	52	52	85	112	177	223	314	717	3617	5427
fastTextRandom	0	10	9	24	28	40	58	60	96	124	195	489	3617	4750
Total TPs per bin	340	156	168	174	208	178	226	230	296	327	419	822	3617	7161

Table 4: Absolute values for disease NER. Best performance in green, worst performance in red, orange when the random initialization is above one of the other initializations.

we respectively note “fastTextPubMed” and “fastTextMimic”. The CharacterBert models are either pre-trained on general corpora (English Wikipedia and OpenWebText (Gokaslan and Cohen, 2019)), which we note “CharBertGen”; or pre-trained on general corpora then re-trained on PubMed and MIMIC-III, which we note “CharBertFromGen”. In all cases the suffix “N2V” refers to a concatenation with the node2vec knowledge representations, with the exception of “Enh.CharBertFromGenN2V” which refers to the variant of CharacterBERT where the node2vec vectors are injected directly within the architecture. This last model is pre-trained on the general corpus then re-trained on PubMed and MIMIC-III in order to be compared with “CharBertFromGen”.

Tables 3 and 4 respectively show absolute values for chemical and disease entity recognition, and Figures 7 and 8 the corresponding bin percentages.

3.2.1 Global performances and pairwise comparison of models

Overall, we can see that the contextual CharacterBert embeddings perform better than the static fastText vectors in both chemical and disease recognition, with the worst performances for randomly initialized fastText embeddings. Moreover, we see that the CharacterBert models trained on medical data perform better than their general versions (Tables 3 and 4, Figures 7 and 8), which confirms the interest of retraining the general models on in-domain data.

Chemical CharacterBert seems to perform rather similarly regardless of the combination with node2vec embeddings. For fastText models, pairwise comparison in Table 5 shows that the introduction of knowledge embeddings (node2vec) improves recall. Comparison of bins further confirms this observation: we can see that the im-

	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	bin-10	bin-11	Recall
EnhancedCharBertFromGenN2V	**29**	62	64	**84**	**79**	83	**85**	91	**95**	95	97	92,83
CharBertFromGen	22	**67**	**67**	79	78	**84**	84	**93**	93	**98**	**99**	**93,14**
CharBertGenN2V	2	**10**	37	**58**	60	**74**	**81**	82	91	95	**99**	**89,62**
CharBertGen	**7**	7	37	56	60	70	79	**84**	**92**	95	97	89,25
fastTextGigawordN2V	**15**	**7**	6	**15**	32	41	53	**67**	**79**	**91**	**99**	**84,29**
fastTextGigaword	0	3	6	10	32	**42**	**54**	43	63	70	93	80,24
fastTextMimicN2V	0	9	**12**	**16**	23	**32**	**44**	**68**	**76**	**92**	**98**	**83,41**
fastTextMimic	**5**	**10**	8	11	**28**	30	43	52	61	84	95	81,15
fastTextPubMedN2V	**10**	**11**	19	**25**	27	47	55	**78**	**82**	**93**	95	84,90
fastTextPubMed	7	10	**26**	21	**44**	**54**	**56**	75	80	92	**98**	**85,74**
fastTextRandomN2V	0	**5**	**10**	**15**	**21**	21	**37**	**43**	**52**	**66**	**90**	**78,03**
fastTextRandom	**2**	2	8	10	16	**22**	30	23	34	29	39	67,0

Table 5: Pairwise comparison of systems with or without addition of Node2Vec embeddings for chemical NER (bin-0 and bin-12 are not considered). The best model for each bin is highlighted in green.

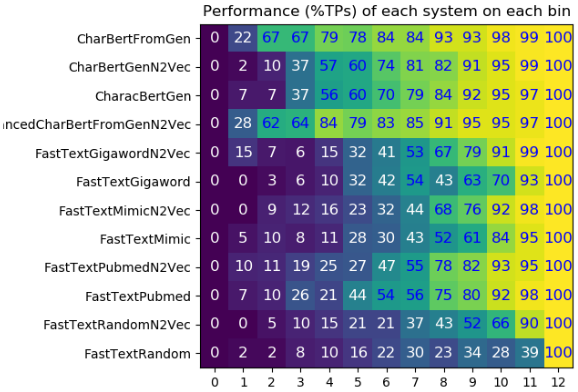


Figure 7: Percentage of labels (true positives) correctly found by each system in each bin for chemical substances.

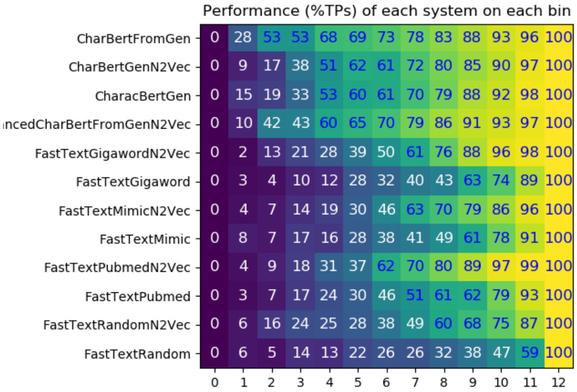


Figure 8: Percentage of labels (true positives) correctly found by each system in each bin for diseases.

provement is made on "easy" entities (bins 8 through 11). However, for "fastTextPubMed" the effect of node2vec is not so clear or even harmful (bins 5 and 6). This phenomenon could be explained by the fact that both PubMed and the BioCreative CDR task are from the biomedical domain while MIMIC-III and Gigaword are from somewhat different domains (clinical and newswire domains respectively). In the case of fastTextPubMed, adding medical knowledge embeddings seems to degrade performance.

Disease While node2vec has a strong positive effect on fastText models regardless of their source corpus, pairwise comparison of recall for disease NER in Table 6 shows that the addition of node2vec is detrimental to CharacterBert models. However, this analysis can be refined by comparing bin-wise performances: for CharacterBert models trained on medical data (top two lines), the versions that do not use node2vec embeddings are better on "more difficult" bins, while the enhanced version are actually better on "easier" bins.

3.2.2 Bin inspection

Browsing through the bins can give an idea of the kinds of entities they hold. This can be done in different ways.

Bin-0 exploration We inspect here the contents of bin-0 for both the chemical and disease recognition tasks, as this bin is supposed to hold false negatives that resist all systems, i.e. the most difficult entities.

Bin-0 for both chemical and disease contains occurrences of abbreviations, which occur quite frequently within parentheses in the context of their full form: for example "bs" for "bile salt" and "rd" (*sic*) for "lenalidomide and dexamethasone" for chemical, "mi" for "myocardial infarction" and "mb" for "microbleeds" for disease. We also spot

	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	bin-10	bin-11	Recall
EnhancedCharBertFromGenN2V	10	42	43	60	65	70	**79**	**86**	**91**	93	**97**	85,14
CharBertFromGen	**28**	**53**	**53**	**68**	**69**	**73**	78	83	88	93	96	**86,11**
CharBertGenN2V	9	17	**38**	51	**62**	61	**72**	**80**	85	90	97	82,87
CharBertGen	**15**	**19**	33	**53**	60	61	70	79	**88**	**92**	**98**	**83,17**
fastTextGigawordN2V	2	**13**	**21**	**28**	**39**	**50**	**61**	**76**	**88**	**96**	**98**	**80,69**
fastTextGigaword	**3**	4	10	12	28	32	40	43	63	74	89	73,40
fastTextMimicN2V	4	7	14	**19**	**30**	**46**	**63**	**70**	**79**	**86**	**96**	**78,40**
fastTextMimic	**8**	7	**17**	16	29	38	41	49	61	78	91	74,71
fastTextPubMedN2V	**4**	**9**	**18**	**31**	**37**	**62**	**70**	**80**	**89**	**97**	**99**	**81,72**
fastTextPubMed	3	7	17	24	30	46	51	61	62	79	93	76,37
fastTextRandomN2V	6	**16**	**24**	**25**	**29**	**38**	**49**	**60**	**68**	**75**	**87**	**75,79**
fastTextRandom	6	5	14	13	22	26	26	32	38	47	59	66,33

Table 6: Pairwise comparison of systems with or without addition of Node2Vec embeddings for disease NER (bin-0 and bin-12 are not considered). The best model for each bin is highlighted in green.

expressions that should perhaps not be in the gold standard, such as “abuse of cocaine and ethanol” tagged as a disease, or typographic errors such as “antithyroidmedications”.

Both bins also hold an important number of single-character tokens such as punctuation marks and digits. For disease recognition, these include the determiner “a”, which occurs most of the time as a part of a multi-word entity. A similar phenomenon occurs with other tokens such as “of”. Occurrences of these words seem to be due to multi-word entities referring to diseases and conditions such as “enlargement of pulse pressure”, “occlusion of renal vessels”, “thrombosis of a normal renal artery”. It seems that multi-word entities account for a significant proportion of the generated errors, where systems only recover the first word of a multi-word entity. For example, chemical bin-0 holds all occurrences of “channel” and “blockers” from “calcium channel blockers”, while occurrences of “calcium” in this context are always labelled correctly.

However, a quick inspection of other bins reveals that those part-of-speech and morphological characteristics (punctuation, single-character entities and abbreviations) are not specific to bin-0. For instance, punctuation marks make for 14% of chemical bin-0 tokens, and for 9 to 28% of bins 1 to 11 (0.07% for bin-12). In the case of disease recognition, punctuation represents 8.8% of bin-0, while ranging from 1.7% to 5.1% of bins 1 to 12 (this difference in proportions between chemical and disease can be explained by the nature of the entities, chemical entities often involving dots or hyphens). Further exploration of the distribution of part-of-speech and morphological categories may lead to some understanding of the bins' contents.

We also found two other phenomena both in chemical bin-0 and in disease bin-0: hapax legomena (‘hapaxes’) and ambiguous tokens.

Hapaxes are tokens that occur only once in the whole data. In bin-0 of the chemical NER task, examples include “adrenergic”, “colony”, “steroidal” or “agents”. In disease bin-0, examples include “bacillary”, “audiogenic”, “choreic”, “teratogenic”.

Ambiguous tokens in bin-0 are due to their multiple or specific meanings in the corpus. This is the case for token “chinese” (note that the corpus is lower-cased), which occurs in “chinese herbal slimming pill”, “chinese herbal”, “chinese herbs”, and is systematically missed in the chemical recognition tasks. We assume that this is probably because it is confused with “chinese” used as the nationality of patients. The same applies to hapax “philadelphia” from “philadelphia chromosome”. These examples lead us to assume that specialized usage of “common” vocabulary terms in chemical or disease entities induces a difficulty for systems.

Distribution across bins Finally, another way to perform bin inspection is to look at the distribution of mentions of a same word across bins. As an illustration, we use the distribution of “calcium” in chemical bins (Table 7): one mention is in bin-1, no mention is in bin-2 and 3, one mention is in bin-4, etc. While most mentions of “calcium” are retrieved by all eleven systems (29 mentions precisely), a total of six of those mentions are individually discovered respectively by exactly one, four, five, nine, nine, and ten systems. This feedback is potentially very useful, since we can then rank every mention in ascending order of difficulty, and proceed to look for explanations for why those six mentions resist detection by a number of systems.

	bin-0	bin-1	bin-2	bin-3	bin-4	bin-5	bin-6	bin-7	bin-8	bin-9	bin-10	bin-11	bin-12
calcium	0	1	0	0	1	1	0	0	0	2	1	0	29

Table 7: Distribution of "calcium" occurrences through bins.

4 Discussion and future work

As we have seen, differential evaluation is a qualitative analysis method that allows for more in-depth evaluation when comparing the behavior of several systems with each other. Rather than relying only on the classical global metrics, it provides an insight into how the performance of each system is actually distributed in automatically-determined subsets of examples relative to other systems, and how systems contribute in their very own way.

As presented in the heatmap we used, harder elements to process are in the first column while easier elements are in the last column. This sorting into several columns allows us to rapidly overview how systems perform on a given task. Based on the analysis we made on the content of bins from several tasks and distinct domains, we observed that the first bin is generally composed of elements such as abbreviations and ambiguous words used in several contexts (some of these contexts are a part of an annotation while other contexts are not); moreover, these elements are often short (two or three characters long), which makes them difficult to process for statistical approaches. In the case of multi-label text classification for Hungarian (Section 3.1.2), differential analysis provided an insight that would have been overlooked by global scores.

Future directions include the following points. First, as we have seen in Section 3.2.2, in the case of named entity recognition, examples composed of several tokens are counted token per token and not as a whole entity. Including this dimension will give another insight into the behavior of models for named-entity recognition. A second direction is to extend the current approach, which focuses on recall, hence true positives against false negatives, to take into account other basic evaluation variables, namely false positives and true negatives. A third useful direction would be to retrieve information on the context of occurrence of examples and their global features: sentence length, direct context, average number of characters per token for each bin, etc. Finally, a fourth direction would be to automatically track the distribution of different mentions of a same word across bins, as we have done manually with "calcium" in the second paragraph of Section 3.2.2. Linked to the previous development regarding contextual information, this would allow us to understand precisely why one particular occurrence of a word is missed while the others are more easily spotted.

Acknowledgments

This work has been made possible by the resources of the MAPA project. The MAPA project is an INEA-funded Action for the European Commission under the Connecting Europe Facility (CEF) – Telecommunications Sector with Grant Agreement No INEA/CEF/ICT/A2019/1927065. https://mapa-project.eu/

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

C.W. Cleverdon. 1960. The ASLIB Cranfield research project on the comparative efficiency of indexing systems. *ASLIB Proceedings*, 12:421–431. ISSN: 0001-253X.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853, Online. Association for Computational Linguistics.

Jonathan G Fiscus. 1997. A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER). In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–357, Santa Barbara, CA.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2007. *English Gigaword, LDC2007T07*. Linguistic Data Consortium, Philadelphia. Web Download.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, page 466–471, USA. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, New York, NY, USA. Association for Computing Machinery.

ICD-10. 2011. *ICD-10. International Statistical Classification of Diseases and Related Health Problems. 10th Revision. Volume 2. Instruction manual*. World Health Organization.

Ridong Jiang, Rafael E. Banchs, and Haizhou Li. 2016. Evaluating and combining name entity recognition systems. In *Proceedings of the Sixth Named Entity Workshop*, pages 21–27, Berlin, Germany. Association for Computational Linguistics.

Alistair E W Johnson, Tom J Pollard, Lu Shen, Li-Wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Sci Data*, 3:160035.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26.

Aurélie Névéol, Aude Robert, Francesco Grippo, Claire Morgand, Chiara Orsi, László Pelikán, Lionel Ramadier, Grégoire Rey, and Pierre Zweigenbaum. 2018. CLEF eHealth 2018 multilingual information extraction task overview: ICD10 coding of death certificates in French, Hungarian and Italian. In *CLEF 2017 Evaluation Labs and Workshop: Online Working Notes*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maxime Peyrard, Wei Zhao, Steffen Eger, and Robert West. 2021. Better than average: Paired evaluation of NLP systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2301–2315, Online. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database J. Biol. Databases Curation*, 2016.