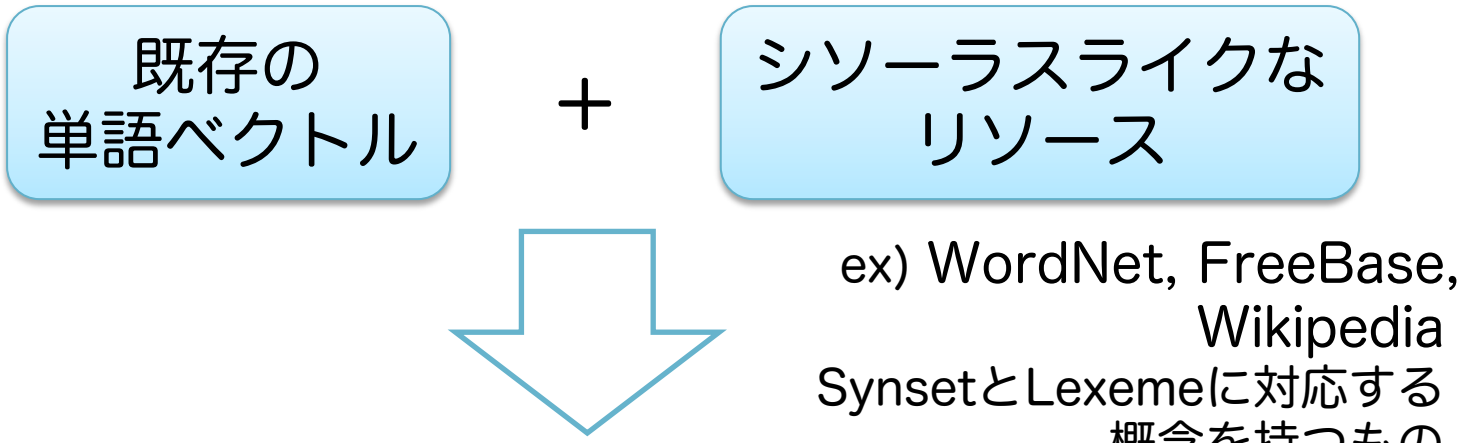
AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes

Sascha Rothe and Hinrich Schütze
University of Munich
In Proc. of ACL 2015

要旨

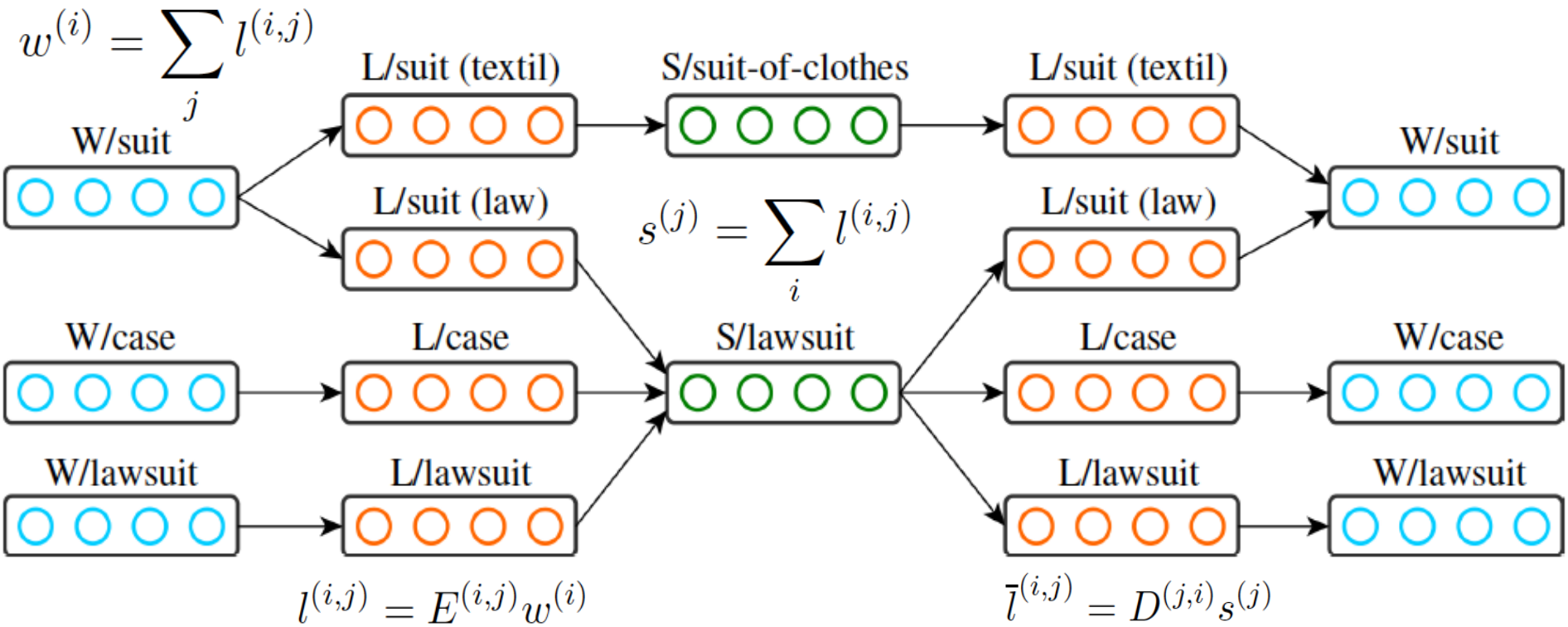
- Synsetと語義(lexeme)のベクトルを学習



入力した単語ベクトルと同じベクトル
空間上の synset・lexeme ベクトル

要旨

- WordNet の word-lexeme, synset-lexeme 関係を利用して autoencoding の枠組みで l と s のベクトルを獲得



このフレームワークの利点

- 単語ベクトルの学習と、シソーラスの利用によるベクトル拡張を切り離している
  - 単語ベクトルはなんでもよく、改良されれば即座に反映可能
- 語義ベクトル学習のために新たに高コストをかけてアノテーションコーパスを用意する必要がない

実験結果の要旨

- WSD
  - シンプルなWSD素性＋獲得したベクトルを利用した素性で、Senseval 2002, 2003 のベストモデルより1%ほどよい結果が得られる。

- コンテキスト依存のWord Similarity
  - Spearman correlation on SCWS
  - Word2Vecオリジナルより良い結果
  - 既存研究と同等かそれ以上のスコア

研究の動機

- The analogy calculus developed by Mikolov et al. (2013a)

King – Man + Woman ≈ Queen

- “The next thing to notice is that this does not only work for words that combine several properties, but also for words that combine several senses.”

Suit ≈ Lawsuit + Business suit

モデル

- 前提となる仮定
  - 単語は lexeme の和からなる
  - Synset は lexeme の和からなる

word: $w^{(i)} \in \mathbb{R}^n$ $\quad w^{(i)} = \sum_j l^{(i,j)}$

synset: $s^{(j)} \in \mathbb{R}^n$

lexeme: $l^{(i,j)} \in \mathbb{R}^n$ $\quad s^{(j)} = \sum_i l^{(i,j)}$

モデル

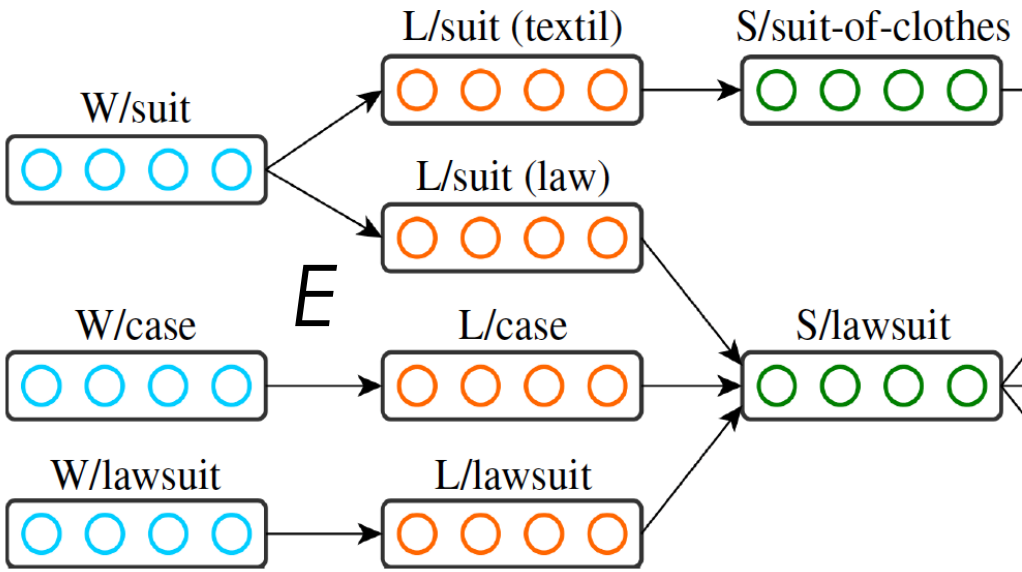
wをlに分解する操作をマトリックスE(i,j) で表現
ただし、次元間の干渉はないと仮定（E(i,j) は対角行列）

$$l^{(i,j)} = E^{(i,j)} w^{(i)}$$

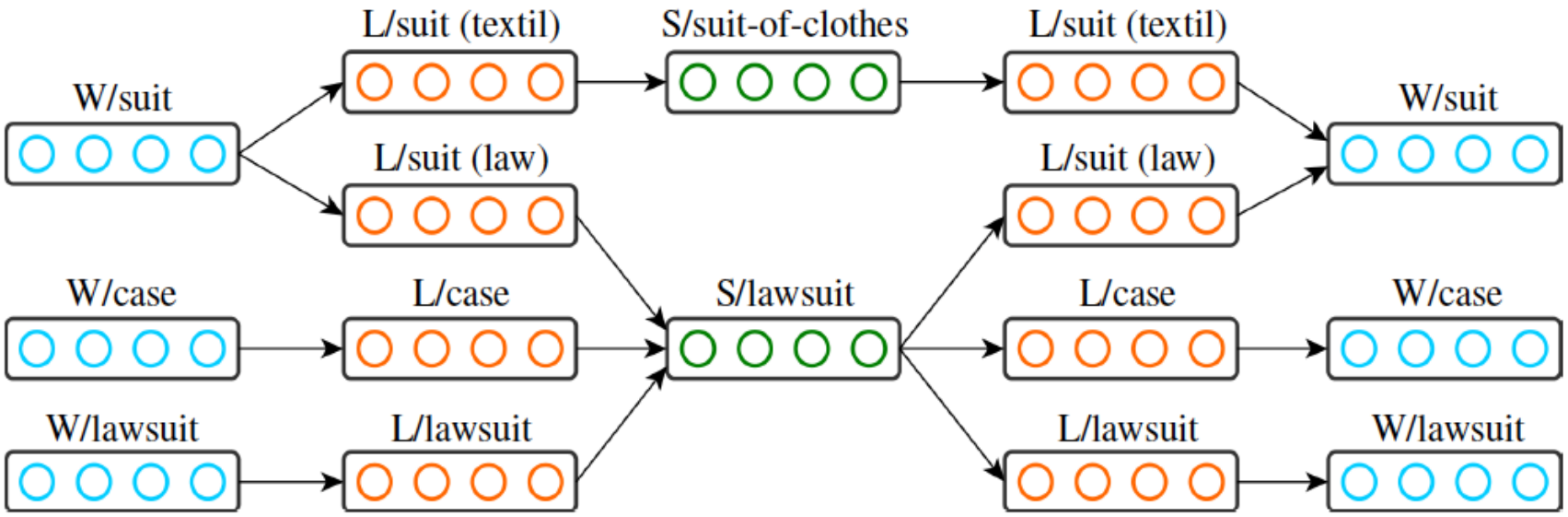
$$\sum_j E^{(i,j)} = I_n \quad \forall i$$

$$w^{(i)} = \sum_j E^{(i,j)} w^{(i)}$$

$$s^{(j)} = \sum_i E^{(i,j)} w^{(i)}$$



学習：autoencoding



$$l^{(i,j)} = E^{(i,j)} w^{(i)}$$

$$w^{(i)} = \sum_j E^{(i,j)} w^{(i)}$$

$$\sum_j E^{(i,j)} = I_n \quad \forall i$$

$$s^{(j)} = \sum_i E^{(i,j)} w^{(i)}$$

$$s^{(j)} = \sum_i D^{(j,i)} s^{(j)}$$

$$\bar{l}^{(i,j)} = D^{(j,i)} s^{(j)}$$

$$\overline{w}^{(i)} = \sum_j D^{(j,i)} s^{(j)}$$

$$\sum_i D^{(j,i)} = I_n \quad \forall j$$

三つの目的関数をバランス

- Synset constraints
  – w と w-barを近づけるように設計
  (オートエンコーダの普通の誤差関数)

- Lexeme constraints
  – l と l-bar を近づけるように設計

- WordNet relation constraints
  – WordNetで関係付けられるsynsetを近づける

$$\alpha \cdot SC + \beta \cdot LC + (1 - \alpha - \beta) \cdot WNC$$

Synset constraints

- E(i,j), D(j,i) は対角行列なので、s や w は次元毎に平行して計算できる

$$\underset{E^{(d)},D^{(d)}}{\operatorname{argmin}} \| D^{(d)} E^{(d)} w^{(d)} - w^{(d)} \| \quad \forall d$$

$w^{(d)}$: $w^{(i)}$ の d 次元目を集めたもの

$s^{(d)}$: $s^{(j)}$ の d 次元目を集めたもの

$E^{(d)} \in \mathbb{R}^{|S| \times |W|}$ ： E(i,j)の d 次元目対角成分を集めたマトリックス

Lexeme constraints

- l と l-bar を近づけるように設計

$$l^{(i,j)} = E^{(i,j)} w^{(i)}$$

$$\bar{l}^{(i,j)} = D^{(j,i)} s^{(j)}$$

$$\underset{E^{(i,j)}, D^{(j,i)}}{\operatorname{argmin}} \left\| E^{(i,j)} w^{(i)} - D^{(j,i)} s^{(j)} \right\| \quad \forall i, j$$

WordNet relation constraints

- lexeme が一つしかない synset をうまく学習するために、 synset 同士で関連があるものを似せる
  - hypernymy, antonymy, similarity, verb group で繋がっているペアを利用

$R \in \mathbb{R}^{r \times |S|}$ ペア数 * synset数のマトリックス
各行は、ペアの片方に対応する列が 1、もう一方が -1

$$\underset{E^{(d)}}{\operatorname{argmin}} \| R \underbrace{E^{(d)} w^{(d)}}_{s^{(d)}} \| \quad \forall d$$

ペアとなる synset の d 次元目が
似ていると 0 に近づく

nearest neighbors of W/suit

S/suit (businessman), L/suit (businessman), L/accomodate, S/suit (be acceptable), L/suit (be acceptable), L/lawsuit, W/lawsuit, S/suit (playing card), L/suit (playing card), S/suit (petition), S/lawsuit, W/countersuit, W/complaint, W/counterclaim

---

nearest neighbors of W/lawsuit

L/lawsuit, S/lawsuit, S/countersuit, L/countersuit, W/countersuit, W/suit, W/counterclaim, S/counterclaim (n), L/counterclaim (n), S/counterclaim (v), L/counterclaim (v), W/sue, S/sue (n), L/sue (n)

---

nearest neighbors of S/suit-of-clothes

L/suit-of-clothes, S/zoot-suit, L/zoot-suit, W/zoot-suit, S/garment, L/garment, S/dress, S/trousers, L/pinstripe, L/shirt, W/tuxedo, W/gabardine, W/tux, W/pinstripe

Figure 2: Five nearest word (W/), lexeme (L/) and synset (S/) neighbors for three items, ordered by cosine

$\alpha = \beta = 0.33$

実験：WSD

			Senseval-2	Senseval-3
IMS feature sets	1	POS	53.6	58.0
	2	surrounding word	57.6	65.3
	3	local collocation	58.7	64.7
	4	S_{naive}-product	56.5	62.2
	5	S-cosine	55.5	60.5
	6	S-product	58.3	64.3
	7	S-raw	56.8	63.1
system comparison	8	MFS	$47.6^{\dagger}$	$55.2^{\dagger}$
	9	Rank 1 system	$64.2^{\dagger}$	72.9
	10	Rank 2 system	$63.8^{\dagger}$	72.6
	11	IMS	$65.2^{\ddagger}$	$72.3^{\ddagger}$
	12	IMS + S_{naive}-prod.	$62.6^{\dagger}$	$69.4^{\dagger}$
	13	IMS + S-cosine	$65.1^{\ddagger}$	$72.4^{\ddagger}$
	14	IMS + S-product	**66.5**	**73.6**
	15	IMS + S-raw	$62.1^{\dagger}$	$66.8^{\dagger}$
	16	IMS + $S_{optimized}$-prod.	*66.6*	*73.6*

The **S-cosine** feature set consists of the k cosines of centroid and synset vectors:

$$< \cos(c, s^{(1)}), \cos(c, s^{(2)}), \ldots, \cos(c, s^{(k)}) >$$

The **S-product** feature set consists of the nk element-wise products of centroid and synset vectors:

$$< c_1 s_1^{(1)}, \ldots, c_n s_n^{(1)}, \ldots, c_1 s_1^{(k)}, \ldots, c_n s_n^{(k)} >$$

where c_i (resp. $s_i^{(j)}$) is element i of c (resp. $s^{(j)}$).

The **S-raw** feature set simply consists of the $n(k+1)$ elements of centroid and synset vectors:

$$< c_1, \ldots, c_n, s_1^{(1)}, \ldots, s_n^{(1)}, \ldots, s_1^{(k)}, \ldots, s_n^{(k)} >$$

$\alpha = 0.2,\ \beta = 0.5$

実験： Word similarity

- SCWS dataset (Huang et al., 2012)
  - 文脈を考慮した単語間の意味類似性評価セット
    - Spearman 相関係数 × 100　　$\alpha = 0.2, \beta = 0.5$

		AvgSim	AvgSimC
1	Huang et al. (2012)	62.8†	65.7†
2	Tian et al. (2014)	–	65.4†
3	Neelakantan et al. (2014)	67.2	69.3
4	Chen et al. (2014)	66.2†	68.9
5	words (word2vec)	66.6‡	66.6†
6	synsets	62.6†	63.7†
7	lexemes	**68.9**	**69.8**

AvgSimC:
文内の単語ベクトル和との類似度で重み付け

実験： Word similarity

- It is interesting that even if we do not take the context into account (method AvgSim) the lexeme embeddings outperform the original word embeddings.

  Replacing a word's embedding by the sum of the embeddings of its senses could generally improve the quality of embeddings (cf. Huang et al. (2012) for a similarpoint).

ット

ベクトル和
で重み付け

1			
2			
3			
4			
5	words (word2vec)	$66.6^{\ddagger}$	$66.6^{\dagger}$
6	synsets	$62.6^{\dagger}$	$63.7^{\dagger}$
7	lexemes	**68.9**	**69.8**

最適なバランスはタスクや素性の組み合わせ方によって異なる

一番良いモデル

embedding 素性のみ

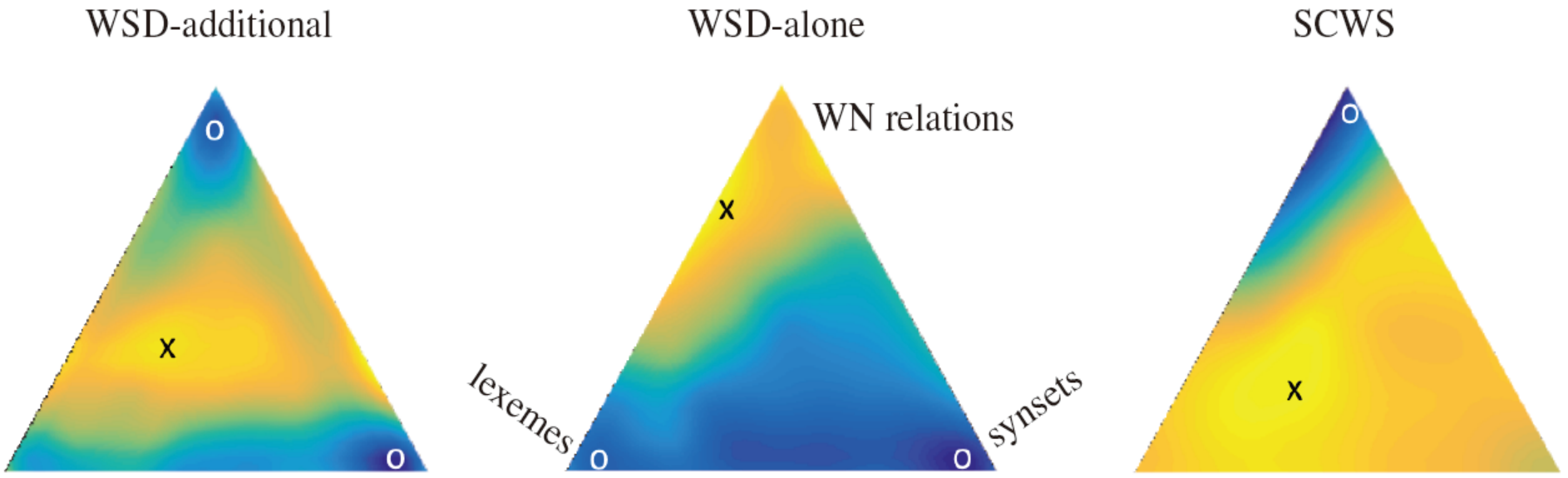


Figure 3: Performance of different weightings of the three constraints (WN relations:top, lexemes:left, synsets:right) on the three tasks WSD-additional, WSD-alone and SCWS. “x” indicates the maximum; “o” indicates a local minimum.

まとめ

- 単語ベクトルからsynsetとlexemeのベクトルを獲得する手法を提案
- 非常に柔軟なモデル：
  - 単語と関係に制約が規定できるデータならば何でも適用可能
- WSDとWord Similarityタスクで最高クラスの精度達成