HAL open science

External knowledge transfer deployment inside a simple double agent Viterbi algorithm

Zied Baklouti

To cite this version:

Zied Baklouti. External knowledge transfer deployment inside a simple double agent Viterbi algorithm. 2021. hal-03358490

HAL Id: hal-03358490

https://hal.science/hal-03358490v1

Preprint submitted on 1 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

External knowledge transfer deployment inside a simple double agent Viterbi algorithm

Zied Baklouti*†

August 2021

Abstract

We consider in this paper deploying external knowledge transfer inside a simple double agent Viterbi algorithm which is an algorithm firstly introduced by the author in his preprint "Hidden Markov Based Mathematical Model dedicated to Extract Ingredients from Recipe Text"(Baklouti, 2021). The key challenge of this work lies in discovering the reason our old model does have bad performances when it is confronted with estimating ingredient state for unknown words and see if deploying external knowledge transfer directly on calculating state matrix could be the solution instead of deploying it only on the back-propagating step.

1 Introduction

Extracting ingredients from a recipe text is a common activity especially for data scientists and developers who want to study recipes or want to make statistical representations about the nutritive values of cuisine recipes. Ingredients are not the only useful information we want to extract, the quantity used for each ingredient and how they are prepared are also interesting information that we can extract by the same method presented in this work. Hidden Markov Models are the first idea that came in my mind because there are previous successful works that used this method for information extraction ((Freitag & McCallum, 2000), (Freitag & McCallum, 1999), (Seymore, McCallum, Rosenfeld, et al., 1999), (Bikel, Miller, Schwartz, & Weischedel, 1998), (Leek, 1997)), and because modeling sequences of words where we must estimate the hidden state is typically a hidden Markov procedure. In this work, we are concentrating on the external knowledge part deployed in what we called a simple double agent Viterbi algorithm.

*Corresponding author: ziedbaklouti@outlook.com

†MSc computer science and mathematics, Paris University

2 Previous work

We used Named Entity recognition as a subtask of information extraction to extract ingredients, HMM are widely used to perform named entity recognition (Morwal, Jahan, & Chopra, 2012) but in its basic technology, HMM is not able to incorporate external knowledge, therefore we developed on this work an advanced Viterbi algorithm to consider POS tags as an external knowledge playing an active role in calculating the state matrix. The major modification made are explained in (Baklouti, 2021).

Extracting ingredients help developers creating recipe recommendation systems, but not only ingredients are targeted in a text recipe, but quantities deployed inside recipes and cooking techniques are also useful information to construct a hybrid recommendation system (Shah, Gaudani, & Balani, 2016) (Ueda, Asanuma, Miyawaki, & Nakajima, 2014) and incorporating POS tags as an external knowledge will help creating more performant hybrid recommendation systems.

Methodologies for transfer learning varies according to the model's task, for example for image classification layers in a neural network may be involved in recognizing specific features and by knowledge transfer, we do not have to retrain those layers in medical images to relearn those specific features (Zhu et al., 2011). Our bigger part of our methodology is explained in (Baklouti, 2021), in this paper we are going to make some modifications in the iteration step to take maximum advantage of already estimated knowledge, in our case already estimated knowledge are POS tags.

3 Methodology

When observing the equations calculating the elements of the state matrix for a second order mono-agent Viterbi algorithm and compare it with the algorithm used in an ingredient extractor (?, ?) , we found an interesting property that kept the second order mono-layered Viterbi more precise in the context of unknown words, in other words when the probability $b_{ij}(v_k) = P(V_r = v_k/\Gamma_r = \gamma_j, T_r = \tau_i)$ is losing its efficiency behind an unknown word , the algorithm used for an ingredient extractor couldn't rely on the other probabilities inside the equations that determines a state matrix. To discover the property in second order Viterbi HMM, we created a table of cases constituted with the list of tokens constituting a sentence and we fill it with the parameters inside the lexical probability and the contextual probability for the most probable cases inside a state matrix δ. For example , for the sentence ' زيت زيتون من النوعية الجيدة' , we want to create its situations table for the token الجيدة, we search first for the most probable cases revealed by the higher values of the state matrix δ, the segment of the matrix δ_6 reveals 3 most probables cases : first case the token الجيدة is not an ingredient , second case the token الجيدة is an ingredient and it is extracted alone, the third case the token الجيدة is a part of an ingredient composed with two words and the word الجيدة is the second word of that ingre-

dient. To calculate one element of a matrix state δ_i for a mono-layered Viterbi algorithm, the algorithm chooses one element of the precedent segment δ_{i-1} of the state matrix and add it with the corresponding lexical probability and transitional probability. We dressed a 3 situations tables for the most probable situations revealed by the higher elements of the state matrix filled with the parameters of those specific probabilities. After reestablishing the situations tables, we can distinguish between two kinds of probabilities:

- **A sane probability**: showing an existing case, for example the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ used in Table 1 calculating $\delta_6(1, 1)$ in equation 2 one of the highest elements of matrix 1 is a sane probability because we are supposing that the token in position $r-1$ is not an ingredient ($\Gamma_{r-1} = 0$) and the estimation obtained from the backtracking step for the token in position r is true ($\Gamma_r = 0$), and the estimation of the ingredient state for the token in position $r-2$ obtained from the mathematical expression $max_{1 \leq i \leq 4}(\delta_5(i, 2) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = \gamma_i))$ in equation 2 is true ($\Gamma_{r-2} = 0$).

- **An insane or a parasite probability**: showing a non-existing case, for example the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, \Gamma_{r-2} = 1)$ in Table 3 used in calculating $\delta_6(1, 3)$ in equation 4 is a parasite probability because we are supposing that the token in position $r - 1$ is the second word of an ingredient ($\Gamma_{r-1} = 2$) and the estimation obtained from the backtracking step for the token in position r is the real value of the token in position r ($\Gamma_r = 0$), and the estimation of the ingredient state for the token in position $r - 2$ obtained from the mathematical expression $max_{1 \leq i \leq 4}(\delta_5(i, 3) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 2, \Gamma_{r-2} = \gamma_i))$ in equation 4 is the real value of the token in position $r - 2$ ($\Gamma_{r-2} = 1$), the first word of an ingredient.

We are supposing that a "good" algorithm is an algorithm where there are the least possible parasite probabilities in its most probable state matrix calculations and we are going to try to ameliorate our ingredient extractor in that way and see if this hypothesis is a valid hypothesis by experimentation.

Estimating an ingredient state using a second order HMM model:
sentence=.,زيت(oil),زيتون(olive),من(of),النوعية(quality),الجيدة(good),.
ingredient state=[0,1,2,0,0,0,0]
ingredient to be extracted=زيت زيتون

$$\delta_6(i,j) = \begin{bmatrix} -41.8209 & -46.5677 & -53.3321 & -54.4383 \\ -41.0199 & -53.0687 & -47.9396 & -54.175 \\ -41.6138 & -53.0402 & -53.4034 & -51.8439 \\ -55.8817 & -56.4674 & -56.1374 & -55.9642 \end{bmatrix} \tag{1}$$

$$\begin{aligned} \delta_6(1,1) &= max_{1\leq i\leq 4}(\delta_5(i,1) + log(a_{i11})) + log(b_{11}(".")) \\ &= max_{1\leq i\leq 4}(\delta_5(i,1) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = \gamma_i))) \\ &+ log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)) \\ &= \underbrace{\delta_5(1,1)}_{-38.13} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0))}_{-0.832} \\ &+ \underbrace{log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0))}_{-2.85} \\ &= -41.82 \end{aligned} \tag{2}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\Gamma_{r-2} = 0$	$\Gamma_{r-1} = 0$	$\Gamma_r = 0$
case2	-	-	-	-	-	$\Gamma_{r-1} = 0$	$\Gamma_r = 0$

Table 1: situations table for $\delta_6(1,1)$ for a second order HMM estimation with all the words are known . The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)$ are describing existing cases from the tagged sentence, they are not parasite probabilities.

$$\begin{aligned} \delta_6(2,1) &= max_{1\leq i\leq 4}(\delta_5(i,2) + log(a_{i21})) + log(b_{21}(".")) \\ &= max_{1\leq i\leq 4}(\delta_5(i,2) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = \gamma_i)) \\ &+ log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)) \\ &= \underbrace{\delta_5(1,2)}_{-38.43} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = 0))}_{-0.351} \\ &+ \underbrace{log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1))}_{-2.238} \\ &= -41.0199 \end{aligned} \tag{3}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\Gamma_{r-2}=0$	$\Gamma_{r-1}=1$	$\Gamma_r=0$
case2	-	-	-	-	-	$\Gamma_{r-1}=1$	$\Gamma_r=0$

Table 2: Situations table for $\delta_6(1,2)$ for a second order HMM estimation. The probabilities $P(\Gamma_r=0/\Gamma_{r-1}=1,\Gamma_{r-2}=0)$ and $P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=1)$ are describing existing cases from the tagged sentence, supposing that the word **الجيدة** is an ingredient . These probabilities are not parasite probabilities.

$$\begin{aligned}
\delta_6(3,1) &= max_{1\leq i\leq 4}(\delta_5(i,3)+log(a_{i31}))+log(b_{31}(\text{"."}))\\
&= max_{1\leq i\leq 4}(\delta_5(i,3)+log(P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=\gamma_i))\\
&+ log(P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2))\\
&= \underbrace{\delta_5(2,3)}_{-38.7}+\underbrace{log(P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=1))}_{-0.043}\\
&+ \underbrace{log(P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2))}_{-2.866}\\
&= -41,61
\end{aligned} \tag{4}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\cancel{\Gamma_{r-2}=1}$	$\Gamma_{r-1}=2$	$\Gamma_r=0$
case2	-	-	-	-	-	$\Gamma_{r-1}=2$	$\Gamma_r=0$

parasite probability

Table 3: Situations table for $\delta_6(3,1)$ for a second order HMM estimation. The probability $P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=1)$ is a parasite probability because $\Gamma_{r-2}=1$ is not the real state of the word **النوعية** and $P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=1)$ is not a parasite probability because it is describing an existing case from the tagged sentence after considering the word **الجيدة** as the second word of an ingredient.

after adding the sentence زيت زيتون من النوعية الجيدة to the dataset:
sentence=.,**زيت**(oil),**زيتون**(olive),**من**(of),**النوعية**(quality),**الجيدة**(good),.
ingredient state=[0,1,2,0,0,0,0]
ingredient to be extracted=**زيت زيتون**

$$\delta_6(i,j)=\begin{bmatrix}-41.0035 & -45.7531 & -53.3305 & -53.6236\\ -40.6117 & -52.6603 & -47.5296 & -53.7664\\ -41.6097 & -53.0374 & -53.4007 & -51.8409\\ -55.7362 & -56.3216 & -55.9917 & -55.8183\end{bmatrix} \tag{5}$$

$$
\begin{aligned}
\delta_6(1,1) &= max_{1\leq i\leq 4}(\delta_5(i,1) + log(a_{i11})) + log(b_{11}(".")) \\
&= max_{1\leq i\leq 4}(\delta_5(i,1) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = \gamma_i))) \\
&+ log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)) \\
&= \underbrace{\delta_5(1,1)}_{-37.314} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0))}_{-0.831} \\
&+ \underbrace{log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0))}_{-2.8567} \\
&= -41.003
\end{aligned} \tag{6}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\Gamma_{r-2} = 0$	$\Gamma_{r-1} = 0$	$\Gamma_r = 0$
case2	-	-	-	-	-	$\Gamma_{r-1} = 0$	$\Gamma_r = 0$

Table 4: situations table for $\delta_6(1,1)$ for a second order HMM estimation with all the words are known after adding the sentence **زيت زيتون من النوعية الجيدة** to the dataset. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)$ are describing existing cases from the tagged sentence, they are not parasite probabilities.

$$
\begin{aligned}
\delta_6(2,1) &= max_{1\leq i\leq 4}(\delta_5(i,2) + log(a_{i21})) + log(b_{21}(".")) \\
&= max_{1\leq i\leq 4}(\delta_5(i,2) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = \gamma_i)) \\
&+ log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)) \\
&= \underbrace{\delta_5(1,2)}_{-38.021} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = 0))}_{-0.3517} \\
&+ \underbrace{log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1))}_{-2.238} \\
&= -40.61
\end{aligned} \tag{7}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\Gamma_{r-2}=0$	$\Gamma_{r-1}=1$	$\Gamma_r=0$
case2	-	-	-	-	-	$\Gamma_{r-1}=1$	$\Gamma_r=0$

Table 5: Situations table for $\delta_6(2,1)$ for a second order HMM estimation with all the words are known after adding the sentence **زيت زيتون من النوعية الجيدة** to the dataset. The probabilities $P(\Gamma_r=0/\Gamma_{r-1}=1,\Gamma_{r-2}=0)$ and $P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=1)$ are describing existing cases from the tagged sentence, supposing that the word **الجيدة** is an ingredient . These probabilities are not parasite probabilities.

$$
\begin{aligned}
\delta_6(3,1) &= max_{1\leq i\leq 4}(\delta_5(i,3)+log(a_{i31}))+log(b_{31}(\text{"."}))\\
&= max_{1\leq i\leq 4}(\delta_5(i,3)+log(P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=\gamma_i))\\
&+ log(P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2))\\
&= \underbrace{\delta_5(2,3)}_{-38.69}+\underbrace{log(P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=1))}_{-0.0437}\\
&+ \underbrace{log(P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2))}_{-2.866}\\
&= -41.609
\end{aligned}
\tag{8}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$\cancel{\Gamma_{r-2}=1}$	$\Gamma_{r-1}=2$	$\Gamma_r=0$	parasite probability
case2	-	-	-	-	-	$\Gamma_{r-1}=2$	$\Gamma_r=0$	

Table 6: Situations table for $\delta_6(3,1)$ for a second order HMM estimation with all the words are known after adding the sentence **زيت زيتون من النوعية الجيدة** to the dataset. The probability $P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=1)$ is a parasite probability because $\Gamma_{r-2}=1$ is not the real state of the word **النوعية** and $P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=1)$ is not a parasite probability because it is describing an existing case from the tagged sentence after considering the word **الجيدة** as the second word of an ingredient.

after adding the sentence كعبة طماطم من الحجم الكبير to the dataset:
sentence=.,زيت(oil),زيتون(olive),من(of),النوعية(quality),الجيدة(good),.
ingredient state=[0,1,2,0,0,0,0]
ingredient to be extracted=زيت زيتون

$$\delta_6(i,j) = \begin{bmatrix} -41.8278 & -46.5781 & -53.3457 & -54.4495 \\ -41.0313 & -53.0804 & -47.9504 & -54.1875 \\ -41.6249 & -53.0533 & -53.4171 & -51.8577 \\ -55.8946 & -56.4807 & -56.1513 & -55.9783 \end{bmatrix} \tag{9}$$

$$\begin{aligned} \delta_6(1,1) &= max_{1\leq i\leq 4}(\delta_5(i,1) + log(a_{i11})) + log(b_{11}(".")) \\ &= max_{1\leq i\leq 4}(\delta_5(i,1) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = \gamma_i))) \\ &+ log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)) \\ &= \underbrace{\delta_5(1,1)}_{-38.138} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0))}_{-0.831} \\ &+ \underbrace{log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0))}_{-2.857} \\ &= -41.827 \end{aligned} \tag{10}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\Gamma_{r-2} = 0$	$\Gamma_{r-1} = 0$	$\Gamma_r = 0$
case2	-	-	-	-	-	$\Gamma_{r-1} = 0$	$\Gamma_r = 0$

Table 7: situations table for $\delta_6(1,1)$ for a second order HMM estimation with all the words are known after adding the sentence كعبة طماطم من الحجم الكبير to the dataset. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)$ are describing existing cases from the tagged sentence, they are not parasite probabilities.

$$\begin{aligned} \delta_6(2,1) &= max_{1\leq i\leq 4}(\delta_5(i,2) + log(a_{i21})) + log(b_{21}(".")) \\ &= max_{1\leq i\leq 4}(\delta_5(i,2) + log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = \gamma_i)) \\ &+ log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)) \\ &= \underbrace{\delta_5(1,2)}_{-38.44} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = 0))}_{-0.3517} \\ &+ \underbrace{log(P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1))}_{-2.239} \\ &= -41.0312 \end{aligned} \tag{11}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\Gamma_{r-2}=0$	$\Gamma_{r-1}=1$	$\Gamma_r=0$
case2	-	-	-	-	-	$\Gamma_{r-1}=1$	$\Gamma_r=0$

Table 8: Situations table for $\delta_6(2,1)$ for a second order HMM estimation with all the words are known after adding the sentence **كعبة طماطم من الحجم الكبير** to the dataset. The probabilities $P(\Gamma_r=0/\Gamma_{r-1}=1,\Gamma_{r-2}=0)$ and $P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=1)$ are describing existing cases from the tagged sentence, supposing that the word **الجيدة** is an ingredient . These probabilities are not parasite probabilities.

$$
\begin{aligned}
\delta_6(3,1) &= max_{1\leq i\leq 4}(\delta_5(i,3)+log(a_{i31}))+log(b_{31}(\text{"."}))\\
&= max_{1\leq i\leq 4}(\delta_5(i,3)+log(P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=\gamma_i))\\
&+ log(P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2))\\
&= \underbrace{\delta_5(2,3)}_{-38.713}+\underbrace{log(P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=1))}_{-0.0437}\\
&+ \underbrace{log(P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2))}_{-2.867}\\
&= -41.6249
\end{aligned}
\tag{12}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$\cancel{\Gamma_{r-2}=1}$	$\Gamma_{r-1}=2$	$\Gamma_r=0$
case2	-	-	-	-	-	$\Gamma_{r-1}=2$	$\Gamma_r=0$

parasite probability

Table 9: Situations table for $\delta_6(3,1)$ for a second order HMM estimation with all the words are known after adding the sentence **كعبة طماطم من الحجم الكبير** to the dataset. The probability $P(\Gamma_r=0/\Gamma_{r-1}=2,\Gamma_{r-2}=1)$ is a parasite probability because $\Gamma_{r-2}=1$ is not the real state of the word **النوعية** and $P(V_r=\text{"."}/\Gamma_r=0,\Gamma_{r-1}=2)$ is not a parasite probability because it is describing an existing case from the tagged sentence after considering the word **الجيدة** as the second word of an ingredient.

For a simple double agent Viterbi algorithm without external knowledge transfer, the parasite probabilities can easily infiltrate the most probable cases inside the equations of a state matrix. For example , for the sentence ' **الجيدة زيت زيتون من النوعية**' , the calculation of the element of the state matrix $\psi_6(7,1)$ in the equation 14 to estimate the ingredient state of the token **الجيدة** shows three parasite probabilities, $P(\Gamma_r=0/\Gamma_{r-1}=0,T_{r-1}=\text{"}$**نعت**$\text{"})$ in table 10 and $P(\Gamma_r=1/\Gamma_{r-1}=0,T_{r-1}=\text{"}$**نعت**$\text{"})$ in table 11 and $P(\Gamma_r=2/\Gamma_{r-1}=3,T_{r-1}=\text{"}$**نعت**$\text{"})$ in table 12 are three parasite probabilities appearing when we

assume that the token **الجيدة** in position r is considered as respectively a non-ingredient $\Gamma_r = 0$ or as an ingredient $\Gamma_r = 1$ or a second word of an ingredient $\Gamma_r = 2$ because the token in position $r - 1$ haven't a POS tag of type **نعت**. To avoid the creation of parasite probability, we decided to use the information obtained in the first layer in calculating the elements of the state matrix, this operation is called external knowledge transfer.

Estimating an ingredient state using a simple double agent Viterbi algorithm without unknown words consideration without external knowledge transfer:

sentence=.,زيت(oil),زيتون(olive),من,النوعية,الجيدة,.

ingredient state=[0,1,2,0,0,0,0]

POS tags=[.,C,D,G,E,F,.] coded as [4,2,3,9,6,7,4]

ingredient to be extracted=زيت زيتون

$$\psi_6(i,j) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 3 & 1 & 3 \\ 2 & 0 & 3 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 1 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix} \tag{13}$$

$$\psi_6(7,1) = argmax_{1\leq i\leq 4}(\delta_5(7,i) + log(a_{7i1})) \tag{14}$$

$$\begin{aligned} \delta_5(7,1) + log(a_{711}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i1})) + \lambda_{max} log(b_{71}(\text{"الجيدة"})) + log(a_{711}) \\ &= \underbrace{\delta_4(7,1)}_{-134.9168} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.717} \\ &+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 0, T_r = \text{"نعت"})}_{-36.629} \\ &+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2,717} \\ &= -176.982 \end{aligned} \tag{15}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	-
case2	-	-	-	-	-	$T_r =$ "نعت" $\Gamma_r = 0$	-
case3	-	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 0$

parasite probability

Table 10: situations table for $\delta_5(7,1)+log(a_{711})$ for a second order HMM estimation, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت"$)$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

$$
\begin{aligned}
\delta_5(7,2) + log(a_{721}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i2})) + \lambda_{max} log(b_{72}(\text{"الجيدة"})) + log(a_{721}) \\
&= \underbrace{\delta_4(7,1)}_{-134.916} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-3.487} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 1, T_r = \text{"نعت"})}_{-38.016} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"نعت"}))}_{-6.583} \\
&= -183
\end{aligned}
\tag{16}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 1$	-
case2	-	-	-	-	-	$T_r =$ "نعت" $\Gamma_r = 1$	-
case3	-	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 1$	$\Gamma_r = 0$

parasite probability

Table 11: situations table for $\delta_5(7,2)+log(a_{721})$ for a second order HMM estimation, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت"$)$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

$$
\begin{aligned}
\delta_5(7,3) + log(a_{731}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i3})) + \lambda_{max} log(b_{73}(\text{"الجيدة"})) + log(a_{731}) \\
&= \underbrace{\delta_4(7,3)}_{-137.789} + \underbrace{log(P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} = \text{"نعت"}))}_{-4.077} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 2, T_r = \text{"نعت"})}_{-37.541} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-0.851} \\
&= -180.259
\end{aligned} \tag{17}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$\cancel{T_{r-1} = \text{"نعت"}}$ $\cancel{\Gamma_{r-1} = 3}$	$\Gamma_r = 2$	-	parasite probability
case2	-	-	-	-	-	$T_r = \text{"نعت"}$ $\Gamma_r = 2$	-	
case3	-	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 2$	$\Gamma_r = 0$	

Table 12: situations table for $\delta_5(7,3)+log(a_{731})$ for a second order HMM estimation, the probability $P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} = \text{"نعت"})$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

after adding the sentence زيت زيتون من النوعية الجيدة to the dataset:
sentence=.,زيت(oil),زيتون(olive),من,النوعية,الجيدة,.
ingredient state=[0,1,2,0,0,0,0]
POS tags=[.,C,D,G,E,F,.] coded as [4,2,3,9,6,7,4]
ingredient to be extracted=زيت زيتون

$$\psi_6(i,j) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 3 & 1 & 3 \\ 2 & 0 & 3 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 1 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix} \tag{18}$$

$$\psi_6(7,1) = argmax_{1\leq i\leq 4}(\delta_5(7,i) + log(a_{7i1}))$$

$$\begin{aligned} \delta_5(7,1) + log(a_{711}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i1})) + \lambda_{max} log(b_{71}(\text{"الجيدة"})) + log(a_{711}) \\ &= \underbrace{\delta_4(7,1)}_{-134.9174} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.7157} \\ &+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 0, T_r = \text{"نعت"})}_{-35.0086} \\ &+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.7157} \\ &= -175.357 \end{aligned} \tag{19}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	-	parasite probability
case2	-	-	-	-	-	$T_r =$ "نعت" $\Gamma_r = 0$	-	
case3	-	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	

Table 13: situations table for $\delta_5(7,1) + log(a_{711})$ for a second order HMM estimation after adding the sentence زيت زيتون من النوعية الجيدة to the dataset , the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت"$)$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

$$
\begin{aligned}
\delta_5(7,2) + log(a_{721}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i2})) + \lambda_{max} log(b_{72}(\text{"الجيدة"})) + log(a_{721}) \\
&= \underbrace{\delta_4(7,1)}_{-134.917} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-3.488} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 1, T_r = \text{"نعت"})}_{-38.0165} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"نعت"}))}_{-6.5843} \\
&= -183.006
\end{aligned}
\tag{20}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 1$	-	parasite probability
case2	-	-	-	-	-	$T_r =$ "نعت" $\Gamma_r = 1$	-	
case3	-	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 1$	$\Gamma_r = 0$	

Table 14: situations table for $\delta_5(7,2)+log(a_{721})$ for a second order HMM estimation after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت"$)$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

$$
\begin{aligned}
\delta_5(7,3) + log(a_{731}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i3})) + \lambda_{max} log(b_{73}(\text{"الجيدة"})) + log(a_{731}) \\
&= \underbrace{\delta_4(7,3)}_{-137.7902} + \underbrace{log(P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} = \text{"نعت"}))}_{-4.0775} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 2, T_r = \text{"نعت"})}_{-37.5415} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-0.8527} \\
&= -180.262
\end{aligned} \tag{21}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$\cancel{T_{r-1} = \text{"نعت"}}$ $\cancel{\Gamma_{r-1} = 3}$	$\Gamma_r = 2$	-	parasite probability
case2	-	-	-	-	-	$T_r = \text{"نعت"}$ $\Gamma_r = 2$	-	
case3	-	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 2$	$\Gamma_r = 0$	

Table 15: situations table for $\delta_5(7,3) + log(a_{731})$ for a second order HMM estimation after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, the probability $P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} = \text{"نعت"})$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت and the ingredient state of the word النوعية is $\Gamma_r = 0$ and not $\Gamma_r = 3$

after adding the sentence كعبة طماطم من الحجم الكبير to the dataset:
sentence=.,زيت(oil),زيتون(olive),من,النوعية,الجيدة,.
ingredient state=[0,1,2,0,0,0,0]
POS tags=[.,C,D,G,E,F,.] coded as [4,2,3,9,6,7,4]
ingredient to be extracted=زيت زيتون

$$\psi_6(i,j) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 3 & 1 & 3 \\ 2 & 0 & 3 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 1 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix} \tag{22}$$

$$\psi_6(7,1) = argmax_{1\leq i\leq 4}(\delta_5(7,i) + log(a_{7i1}))$$

$$\begin{aligned} \delta_5(7,1) + log(a_{711}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i1})) + \lambda_{max} log(b_{71}(\text{"الجيدة"})) + log(a_{711}) \\ &= \underbrace{\delta_4(7,1)}_{-134.9407} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.7157} \\ &+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 0, T_r = \text{"نعت"})}_{-36.6363} \\ &+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.7157} \\ &= -177.008 \end{aligned} \tag{23}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	-	parasite probability
case2	-	-	-	-	-	$T_r =$ "نعت" $\Gamma_r = 0$	-	
case3	-	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	

Table 16: situations table for $\delta_5(7,1)+log(a_{711})$ for a second order HMM estimation after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت"$)$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

$$
\begin{aligned}
\delta_5(7,2) + log(a_{721}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i2})) + \lambda_{max} log(b_{72}(\text{"الجيدة"})) + log(a_{721}) \\
&= \underbrace{\delta_4(7,1)}_{-134.9407} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-3.488} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 1, T_r = \text{"نعت"})}_{-38.02485} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"نعت"}))}_{-6.5843} \\
&= -183.0379
\end{aligned}
\tag{24}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 0$	$\Gamma_r = 1$	-	parasite probability
case2	-	-	-	-	-	$T_r =$ "نعت" $\Gamma_r = 1$	-	
case3	-	-	-	-	-	$T_{r-1} =$ "نعت" $\Gamma_{r-1} = 1$	$\Gamma_r = 0$	

Table 17: situations table for $\delta_5(7,2)+log(a_{721})$ for a second order HMM estimation after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت"$)$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

$$
\begin{aligned}
\delta_5(7,3) + log(a_{731}) &= max_{1\leq i\leq 4}(\delta_4(7,i) + log(a_{7i3})) + \lambda_{max} log(b_{73}(\text{"الجيدة"})) + log(a_{731}) \\
&= \underbrace{\delta_4(7,3)}_{-137.8215} + \underbrace{log(P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} = \text{"نعت"}))}_{-4.0775} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 2, T_r = \text{"نعت"})}_{-37.5509} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-0.85277} \\
&= -180.30279
\end{aligned}
\tag{25}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$\cancel{T_{r-1} = \text{"نعت"}}$ $\cancel{\Gamma_{r-1} = 3}$	$\Gamma_r = 2$	-	parasite probability
case2	-	-	-	-	-	$T_r = \text{"نعت"}$ $\Gamma_r = 2$	-	
case3	-	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 2$	$\Gamma_r = 0$	

Table 18: situations table for $\delta_5(7,3)+log(a_{731})$ for a second order HMM estimation after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"})$ is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت

The modifications made to ameliorate the performances of a simple double agent Viterbi algorithm are:

- Inside the mathematical expression $max_j[\delta_{l-1}(i,j) + \log(a_{ijk})]$, a_{ijk} is replaced by $a_{y(l)jk}$ where $y(l)$ is the POS tag observed in the corresponding sequence of POS tags at position l.
- Inside the mathematical expression $max_j[\delta_{l-1}(i,j) + \log(a_{ijk})]$, $\delta_{l-1}(i,j)$ is replaced by $\delta_{l-1}(y(l),j)$ where $y(l)$ is the POS tag observed in the corresponding sequence of POS tags at position l.

We observe a tremendous decrease of the parasite probabilities infiltrated inside the equations of the most probable cases of a state matrix element. For example, the parasite probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"})$ in in Table 11 is eliminated and replaced by the sane probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"منعوت"})$ in in Table 20 by external knowledge transfer consisting of introducing the POS tag estimated in first layer at position $r-1$.

Estimating an ingredient state using a double agent Viterbi algorithm with external knowledge transfer:

sentence=.,زيت(oil),زيتون(olive),من,النوعية,الجيدة,.

ingredient state=[0,1,2,0,0,0,0]

POS tags=[.,C,D,G,E,F,.] coded as [4,2,3,9,6,7,4]

ingredient to be extracted=زيت زيتون

$$
\begin{aligned}
\delta_5(7,1) + log(a_{711}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i1})) + \lambda_{max} log(b_{71}(\text{"الجيدة"})) + log(a_{711}) \\
&= \underbrace{\delta_4(6,2)}_{-94.593} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"منعوت"}))}_{-2.189} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 0, T_r = \text{"نعت"})}_{-36.63} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.7178} \\
&= -136.13
\end{aligned}
\tag{26}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$T_{r-1} = \text{"منعوت"}$ $\cancel{\Gamma_{r-1} = 1}$	$\Gamma_r = 0$	-
case2	-	-	-	-	-	$T_r = \text{"نعت"}$ $\Gamma_r = 0$	-
case3	-	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$

parasite probability

Table 19: situations table for $\delta_5(7,1) + log(a_{711})$ for a double agent Viterbi with external knowledge transfer, $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"منعوت"})$ is a parasite probability because the ingredient state of the word النوعية is $\Gamma_{r-1} = 0$ and not $\Gamma_{r-1} = 1$

$$
\begin{aligned}
\delta_5(7,2) + log(a_{721}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i2})) + \lambda_{max} log(b_{72}(\text{"الجيدة"})) + log(a_{721}) \\
&= \underbrace{\delta_4(6,1)}_{-93.928} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"منعوت"}))}_{-7.559} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 1, T_r = \text{"نعت"})}_{-38.015} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"نعت"}))}_{-6.583} \\
&= -146.087
\end{aligned}
\tag{27}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1}=$ "منعوت" $\Gamma_{r-1}=0$	$\Gamma_r=1$	-	parasite probability eliminated
case2	-	-	-	-	-	$T_r=$ "نعت" $\Gamma_r=1$	-	
case3	-	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=1$	$\Gamma_r=0$	

Table 20: situations table for $\delta_5(7,2)+log(a_{721})$ for a double agent Viterbi with external knowledge transfer, $P(\Gamma_r=1/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت"$)$ is a sain probability because the word النوعية is a منعوت and it is not an ingredient.

$$\begin{aligned}
\delta_5(7,3)+log(a_{731}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i)+log(a_{POS[5]i3}))+\lambda_{max}log(b_{73}(\text{"الجيدة"}))+log(a_{731}) \\
&= \underbrace{\delta_4(6,2)}_{-94.59}+\underbrace{log(P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=\text{"منعوت"}))}_{-2.029} \\
&+\underbrace{\lambda_{max}log(P(V_r=\text{"الجيدة"}/\Gamma_r=2, T_r=\text{"نعت"})}_{-37.541} \\
&+\underbrace{log(P(\Gamma_r=0/\Gamma_{r-1}=2, T_{r-1}=\text{"نعت"}))}_{-0.851} \\
&= -135.015
\end{aligned} \tag{28}$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1}=$ "منعوت" $\cancel{\Gamma_{r-1}=1}$	$\Gamma_r=2$	-	parasite probability maintained
case2	-	-	-	-	-	$T_r=$ "نعت" $\Gamma_r=2$	-	
case3	-	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=2$	$\Gamma_r=0$	

Table 21: situations table for $\delta_5(7,3)+log(a_{731})$ for a double agent Viterbi with external knowledge transfer, $P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت"$)$ is a parasite probability because the word النوعية is not an ingredient.

after adding the sentence كعبة طماطم من الحجم الكبير to the dataset:
sentence=.,زيت(oil),زيتون(olive),من,النوعية,الجيدة,.
ingredient state=[0,1,2,0,0,0,0]
POS tags=[.,C,D,G,E,F,.] coded as [4,2,3,9,6,7,4]
ingredient to be extracted=زيت زيتون

$$
\begin{aligned}
\delta_5(7,1) + log(a_{711}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i1})) + \lambda_{max} log(b_{71}(\text{"الجيدة"})) + log(a_{711}) \\
&= \underbrace{\delta_4(6,2)}_{-94.59} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"منعوت"}))}_{-2.189} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 0, T_r = \text{"نعت"})}_{-36.637} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.715} \\
&= -136.133
\end{aligned} \tag{29}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.
case1	-	-	-	-	$T_{r-1} = \text{"منعوت"}$ $\cancel{\Gamma_{r-1} = 1}$	$\Gamma_r = 0$	-
case2	-	-	-	-	-	$T_r = \text{"نعت"}$ $\Gamma_r = 0$	-
case3	-	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$

parasite probability

Table 22: situations table for $\delta_5(7,1) + log(a_{711})$ for a double agent Viterbi with external knowledge transfer, after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"منعوت"})$ is a parasite probability because the word النوعية is not an ingredient in reality given الجيدة have an ingredient state $\Gamma_r = 0$.

$$
\begin{aligned}
\delta_5(7,2) + log(a_{721}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i2})) + \lambda_{max} log(b_{72}(\text{"الجيدة"})) + log(a_{721}) \\
&= \underbrace{\delta_4(6,1)}_{-93.91} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"منعوت"}))}_{-7.56} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 1, T_r = \text{"نعت"})}_{-38.024} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"نعت"}))}_{-6.584} \\
&= -146.078
\end{aligned} \tag{30}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1}=$ "منعوت" $\Gamma_{r-1}=0$	$\Gamma_r=1$	-	saine probability
case2	-	-	-	-	-	$T_r=$ "نعت" $\Gamma_r=1$	-	
case3	-	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=1$	$\Gamma_r=0$	

Table 23: situations table for $\delta_5(7,2)+log(a_{721})$ for a double agent Viterbi with external knowledge transfer, after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, $P(\Gamma_r=0/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت"$)$ is not a parasite probability because the word النوعية is not an ingredient and النوعية POS tag is منعوت .

$$
\begin{aligned}
\delta_5(7,3)+log(a_{731}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i)+log(a_{POS[5]i3}))+\lambda_{max}log(b_{73}(\text{"الجيدة"}))+log(a_{731}) \\
&= \underbrace{\delta_4(6,2)}_{-94.59}+\underbrace{log(P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=\text{"منعوت"}))}_{-2.03} \\
&+\underbrace{\lambda_{max}log(P(V_r=\text{"الجيدة"}/\Gamma_r=2, T_r=\text{"نعت"})}_{-37.55} \\
&+\underbrace{log(P(\Gamma_r=0/\Gamma_{r-1}=2, T_{r-1}=\text{"نعت"}))}_{-0.852} \\
&= -135.02
\end{aligned}
\tag{31}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1}=$ "منعوت" $\cancel{\Gamma_{r-1}=1}$	$\Gamma_r=2$	-	parasite probability
case2	-	-	-	-	-	$T_r=$ "نعت" $\Gamma_r=2$	-	
case3	-	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=2$	$\Gamma_r=0$	

Table 24: situations table for $\delta_5(7,3)+log(a_{731})$ for a double agent Viterbi with external knowledge transfer, after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, $P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت"$)$ is a parasite probability because the word النوعية have an ingredient state $\Gamma_{r-1}=0$ and not $\Gamma_{r-1}=1$.

after adding the sentence زيت زيتون من النوعية الجيدة to the dataset:
sentence=.,زيت(oil),زيتون(olive),من,النوعية,الجيدة,.

ingredient state=[0,1,2,0,0,0,0]
POS tags=[.,C,D,G,E,F,.] coded as [4,2,3,9,6,7,4]
ingredient to be extracted=زيت زيتون

$$
\begin{aligned}
\delta_5(7,1) + log(a_{711}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i1})) + \lambda_{max} log(b_{71}(\text{"الجيدة"})) + log(a_{711}) \\
&= \underbrace{\delta_4(6,1)}_{-92.101} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"منعوت"}))}_{-3.2605} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 0, T_r = \text{"نعت"})}_{-35.009} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"نعت"}))}_{-2.7159} \\
&= -133.08
\end{aligned}
\tag{32}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1} = \text{"منعوت"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	-	
case2	-	-	-	-	-	$T_r = \text{"نعت"}$ $\Gamma_r = 0$	-	parasite probability eliminated
case3	-	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	

Table 25: situations table for $\delta_5(7,1) + log(a_{711})$ for a double agent Viterbi with external knowledge transfer after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"منعوت"})$ is not a parasite probability because the word النوعية is not an ingredient and the word النوعية is منعوت given the word الجيدة is not an ingredient.

$$
\begin{aligned}
\delta_5(7,2) + log(a_{721}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i2})) + \lambda_{max} log(b_{72}(\text{"الجيدة"})) + log(a_{721}) \\
&= \underbrace{\delta_4(6,1)}_{-92.101} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} = \text{"منعوت"}))}_{-7.56} \\
&+ \underbrace{\lambda_{max} log(P(V_r = \text{"الجيدة"}/\Gamma_r = 1, T_r = \text{"نعت"})}_{-38.016} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"نعت"}))}_{-6.584} \\
&= -144.26
\end{aligned}
\tag{33}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1}=$ "منعوت" $\Gamma_{r-1}=0$	$\Gamma_r=1$	-	parasite probability eliminated
case2	-	-	-	-	-	$T_r=$ "نعت" $\Gamma_r=1$	-	
case3	-	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=1$	$\Gamma_r=0$	

Table 26: situations table for $\delta_5(7,2)+log(a_{721})$ for a double agent Viterbi with external knowledge transfer after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, $P(\Gamma_r=1/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت"$)$ is not a parasite probability because the word النوعية is not an ingredient and the word النوعية is منعوت given the word الجيدة is not an ingredient.

$$
\begin{aligned}
\delta_5(7,3)+log(a_{731}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i)+log(a_{POS[5]i3}))+\lambda_{max}log(b_{73}(\text{"الجيدة"}))+log(a_{731}) \\
&= \underbrace{\delta_4(6,2)}_{-94.401}+\underbrace{log(P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=\text{"منعوت"}))}_{-2.03} \\
&+\underbrace{\lambda_{max}log(P(V_r=\text{"الجيدة"}/\Gamma_r=2, T_r=\text{"نعت"})}_{-37.541} \\
&+\underbrace{log(P(\Gamma_r=0/\Gamma_{r-1}=2, T_{r-1}=\text{"نعت"}))}_{-0.852} \\
&= -134.826
\end{aligned}
\tag{34}
$$

	.	زيت	زيتون	من	النوعية	الجيدة	.	
case1	-	-	-	-	$T_{r-1}=$ "منعوت" $\cancel{\Gamma_{r-1}=1}$	$\Gamma_r=2$	-	parasite probability
case2	-	-	-	-	-	$T_r=$ "نعت" $\Gamma_r=2$	-	
case3	-	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=2$	$\Gamma_r=0$	

Table 27: situations table for $\delta_5(7,3)+log(a_{731})$ for a double agent Viterbi with external knowledge transfer after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, $P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت"$)$ is a parasite probability because the word النوعية is not an ingredient in reality.

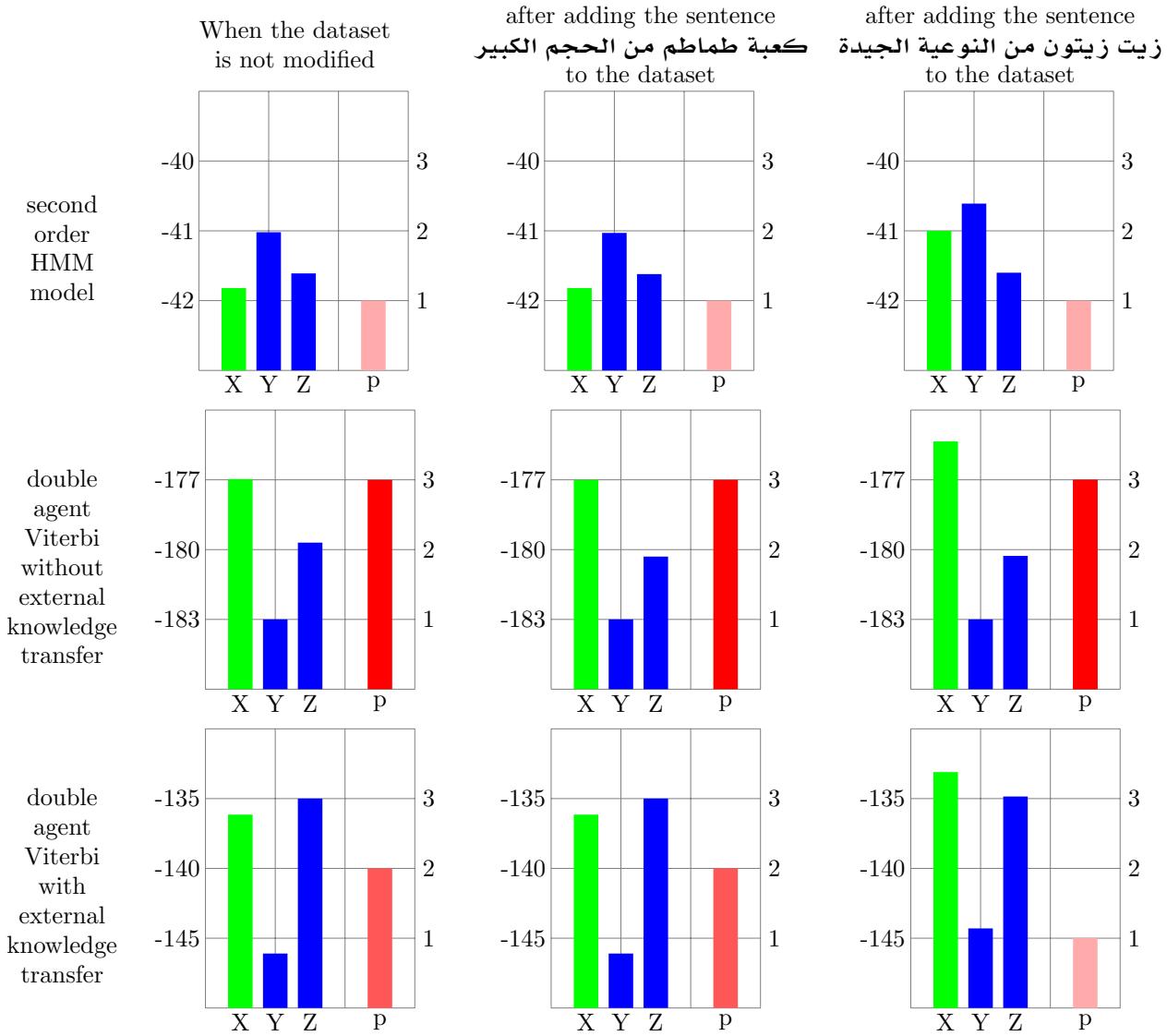


Figure 1: data sensitivity bar plot for the word الجيدة inside the sentence زيت زيتون من النوعية الجيدة under graduated conditions. X is the first most probable case inside a state matrix الجيدة is not an ingredient), Y is the second most probable case inside a state matrix الجيدة is an ingredient constituted with a single word) and Z is the third most probable case inside a state matrix (الجيدة is the second word of an ingredient constituted with two elements) , P is the number of parasite probabilities observed when calculating X and Y and Z.

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	number of known words
A (إسم)	98.47 %	65.57 %	NULL	98.47 %	0	1112
B (رقم)	99.79 %	74.95 %	NULL	99.79 %	0	947
C (إسم معرف)	96.99 %	83.79 %	NULL	96.99 %	0	798
D (إضافة)	94.69 %	70.83 %	NULL	94.69 %	0	791
E (منعوت)	98.8 %	87.95 %	NULL	98.8 %	0	750
F (نعت)	85.31 %	69.92 %	NULL	85.31 %	0	844
G (حرف جر)	99.54 %	49.89 %	NULL	99.54 %	0	219
H (إسم مجرور)	86.81 %	60.21 %	NULL	86.81 %	0	182
I (وحدة قيس)	99.36 %	49.89 %	NULL	99.36 %	0	469
J (واو العطف)	98.77 %	49.69 %	NULL	98.77 %	0	162
K (فعل مبني للمجهول)	78.57 %	31.45 %	NULL	78.57 %	0	140
L (المفعول المطلق)	60 %	25 %	NULL	60 %	0	5
M (أداةُ عَطْفٌ غير واو العطف)	92.31 %	48 %	NULL	92.31 %	0	39
.	99.9 %	49.97 %	NULL	99.9 %	0	1972

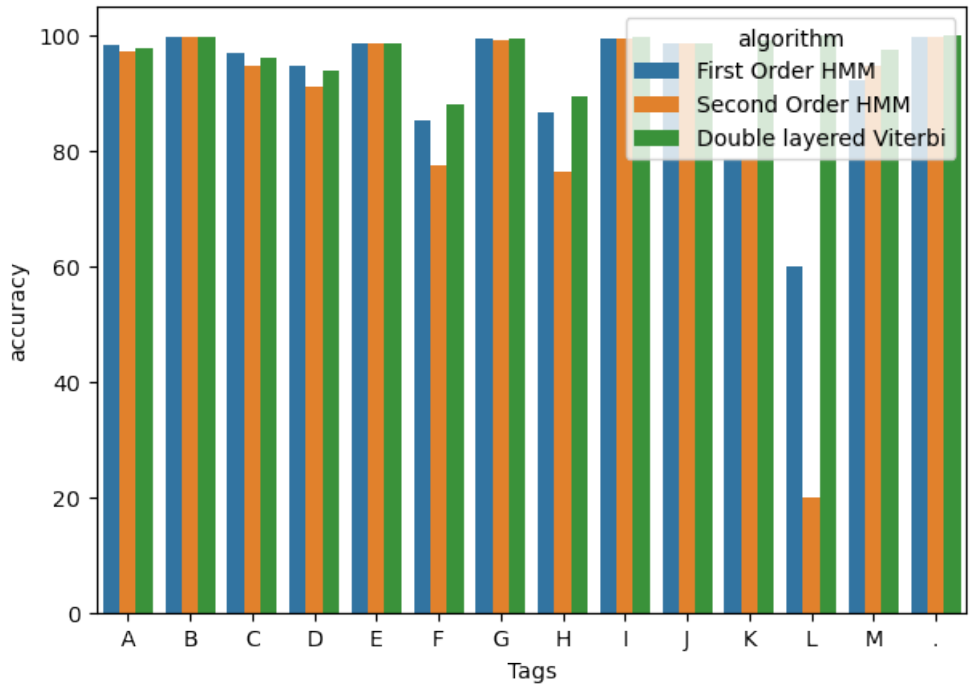
Table 28: Performances for each tag of a first order HMM , the trained dataset is used as a test dataset , and there is zero unknown words to predict

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	number of known words
A (إسم)	97.3 %	64.68 %	NULL	97.3 %	0	1112
B (رقم)	99.89 %	83.31 %	NULL	99.89 %	0	947
C (إسم معرف)	94.86 %	85.36 %	NULL	94.86 %	0	798
D (إضافة)	91.02 %	68.01 %	NULL	91.02 %	0	791
E (منعوت)	98.53 %	98.84 %	NULL	98.53 %	0	750
F (نعت)	77.61 %	54.6 %	NULL	77.61 %	0	844
G (حرف جر)	99.09 %	49.77 %	NULL	99.09 %	0	219
H (إسم مجرور)	76.37 %	52.49 %	NULL	76.37 %	0	182
I (وحدة قيس)	99.57 %	74.89 %	NULL	99.57 %	0	469
J (واو العطف)	98.77 %	49.69 %	NULL	98.77 %	0	162
K (فعل مبني للمجهول)	78.57 %	32.2 %	NULL	78.57 %	0	140
L (المفعول المطلق)	20 %	16.67 %	NULL	20 %	0	5
M (أداةُ عَطْفٌ غير واو العطف)	94.87 %	48.68 %	NULL	94.87 %	0	39
.	99.85 %	49.96 %	NULL	99.85 %	0	3945

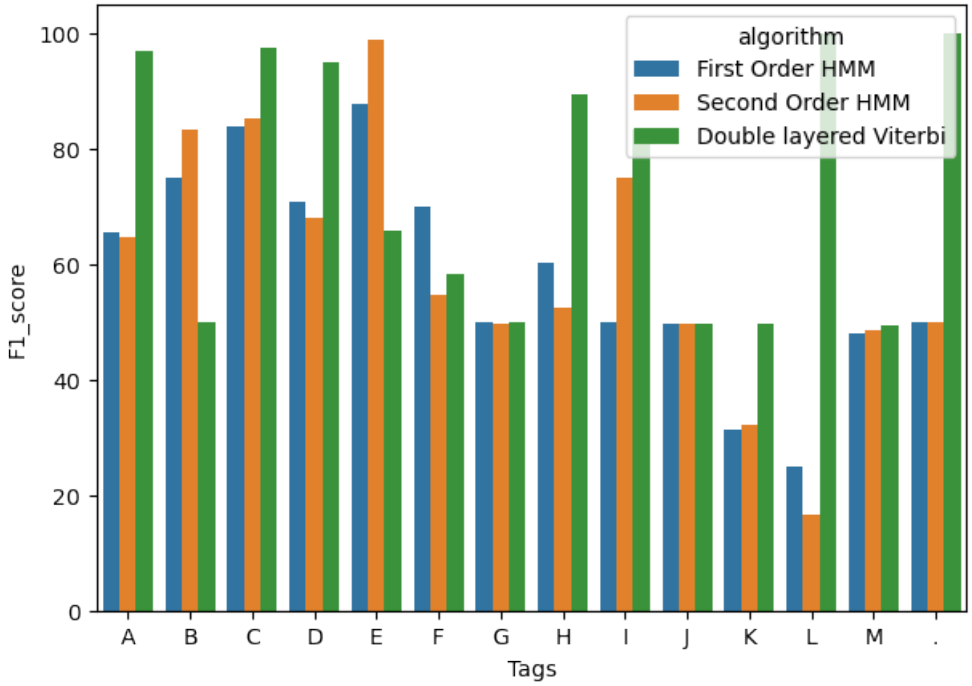
Table 29: Performances for each tag of a second order HMM , the trained dataset is used as a test dataset , and there is zero unknown words to predict

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	number of known words
A (إسم)	97.84 %	97.06 %	NULL	97.84 %	0	1113
B (رقم)	99.79 %	49.95 %	NULL	99.79 %	0	947
C (إسم معرف)	96.25 %	97.44 %	NULL	96.25 %	0	799
D (إضافة)	93.94 %	94.92 %	NULL	93.94 %	0	792
E (منعوت)	98.8 %	65.77 %	NULL	98.8 %	0	748
F (نعت)	88.02 %	58.34 %	NULL	88.02 %	0	843
G (حرف جر)	99.54 %	49.89 %	NULL	99.54 %	0	219
H (إسم مجرور)	89.56 %	89.36 %	NULL	89.56 %	0	182
I (وحدة قيس)	99.79 %	83.28 %	NULL	99.79 %	0	469
J (واو العطف)	98.77 %	49.69 %	NULL	98.77 %	0	162
K (فعل مبني للمجهول)	99.29 %	49.82 %	NULL	99.29 %	0	140
L (المفعول المطلق)	100 %	100 %	NULL	100 %	0	5
M (أداةُ عَطْفٌ غير واو العطف)	97.44 %	49.35 %	NULL	97.44 %	0	39
.	100 %	100 %	NULL	100 %	0	3946

Table 30: Performance comparison using a simple double agent Viterbi with external knowledge transfer with 100 % accuracy on first layer , the trained dataset is used as a test dataset to avoid unknown words prediction



(a) accuracy for each tag without unknown words consideration



(b) f1 score for each tag without unknown words consideration

Figure 2: Performance comparison between a double agent Viterbi algorithm with external knowledge transfer and other mono layered Viterbi algorithm

	Accuracy	F1 score
First order Hidden Markov Model applied to tokens to predict ingredients	96.61 %	70.06 %
Second order Hidden Markov Model applied to tokens to predict ingredients	95.82 %	68.18 %
Our ingredient extractor with 100 % accuracy on the first layer with $\lambda = 4$	98.43 %	81.43 %
A double layered Viterbi algorithm with 100 % accuracy on the first layer with $\lambda = 4$	97.7 %	96.32 %

Table 31: Comparison of performances between a simple double agent Viterbi algorithm with external knowledge transfer and other HMM mono-agent methods using the trained corpus as a testing dataset

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	unknown words percentage
0	94.77 %	67.67 %	67.78 %	96.33 %	78	9.67 %
1	92.96 %	65.52 %	63.93 %	95.14 %	107	13.26 %
2	94.03 %	67.36 %	69.9 %	95.68 %	88	10.9 %
3	93.95 %	87.84 %	62.65 %	95.64 %	75	9.29 %
4	92.91 %	87.29 %	66.99 %	94.54 %	92	11.4 %
5	95.78 %	69.61 %	76.4 %	96.83 %	83	10.29 %
6	94.38 %	67.53 %	54.65 %	96.49 %	80	9.91 %
7	95.08 %	91.17 %	61.22 %	97.17 %	90	11.15 %
8	96.83 %	94.04 %	79.76 %	97.7 %	77	9.54 %
9	94.92 %	90.7 %	64.95 %	96.72 %	88	10.9 %
Avg	94.56 %	78.87 %	66.82 %	96.22 %	86	10.63 %

Table 32: Performances of first order HMM used to predict ingredients state using 10 fold cross-validation with unknown word consideration, the test dataset have 20 % sentences from the total dataset

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	unknown words percentage
0	93.72 %	64.13 %	70.0 %	94.81 %	78	9.67 %
1	92.48 %	62.96 %	64.75 %	94.14 %	107	13.26 %
2	94.04 %	65.53 %	70.87 %	95.29 %	88	10.9 %
3	93.89 %	86.87 %	68.67 %	94.97 %	75	9.29 %
4	93.42 %	86.61 %	66.02 %	94.81 %	92	11.4 %
5	93.83 %	65.85 %	62.92 %	95.18 %	83	10.29 %
6	93.76 %	65.63 %	60.47 %	95.19 %	80	9.91 %
7	94.04 %	88.56 %	55.1 %	95.96 %	90	11.15 %
8	95.34 %	90.99 %	60.71 %	96.77 %	77	9.54 %
9	94.34 %	88.56 %	61.86 %	95.91 %	88	10.9 %
Avg	93.89 %	76.57 %	64.14 %	95.3 %	86	10.63 %

Table 33: Performances of second order HMM used to predict ingredients state using 10 fold cross-validation with unknown word consideration, the test dataset have 20 % sentences from the total dataset

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	unknown words percentage
0	95.78 %	67.91 %	61.11 %	97.38 %	78	9.67 %
1	94.34 %	66.51 %	68.85 %	95.87 %	107	13.26 %
2	94.98 %	67.32 %	67.96 %	96.44 %	88	10.9 %
3	95.83 %	67.71 %	62.65 %	97.25 %	75	9.29 %
4	94.88 %	67.76 %	71.84 %	96.05 %	92	11.4 %
5	95.9 %	69.57 %	70.79 %	97.0 %	83	10.29 %
6	94.43 %	65.92 %	55.81 %	96.08 %	80	9.91 %
7	94.04 %	65.54 %	51.02 %	96.17 %	90	11.15 %
8	95.63 %	68.82 %	60.71 %	97.06 %	77	9.54 %
9	94.39 %	65.95 %	56.7 %	96.21 %	88	10.9%
Avg	95.02 %	67.3 %	62.74 %	96.55 %	86	10.63 %

Table 34: Average performances in predicting ingredients states with $\lambda = 4$ when accuracy is 100% on first layer with our ingredient extractor

tag	accuracy	f1 score	accuracy for unknown words	accuracy for known words	number of unknown words	unknown words percentage
0	96.61 %	69.39 %	80 %	97.38 %	78	9.67 %
1	94.89 %	67.6 %	77.05 %	95.96 %	107	13.26 %
2	95.13 %	67.76 %	85.44 %	95.65 %	88	10.9 %
3	95.87 %	91.04 %	80.72 %	96.53 %	75	9.29 %
4	95.49 %	90.68 %	78.64 %	96.34 %	92	11.4 %
5	95.81 %	69.1 %	74.16 %	97.0 %	83	10.29 %
6	94.43 %	65.92 %	55.81 %	96.08 %	80	9.91 %
7	94.04 %	65.54 %	51.02 %	96.17 %	90	11.15 %
8	95.63 %	68.82 %	60.71 %	97.06 %	77	9.54 %
9	94.39 %	65.95 %	56.7 %	96.21 %	88	10.9%
Avg	95.24 %	72.19 %	70.13 %	96.42 %	86	10.63 %

Table 35: Average performances in predicting ingredients states with $\lambda = 4$ when accuracy is 100% on first layer using a double layered Viterbi Algorithm

	known	unknown	overall	F1 score
First order Hidden Markov Model applied to tokens	96.22 %	66.82 %	94.56 %	78.87 %
Second order Hidden Markov Model applied to tokens	95.3 %	64.14 %	93.89 %	76.57 %
Our ingredient extractor with 100 % accuracy on the first layer with $\lambda = 4$	96.55 %	62.74 %	95.02 %	67.3 %
A double layered Viterbi algorithm with 100 % accuracy on the first layer with $\lambda = 4$	96.42 %	70.13 %	95.24 %	72.19 %

Table 36: Comparison of performances between the 4 methods using 20 % of the total dataset as a test dataset

Estimating an ingredient state using a simple double agent Viterbi without unknown words consideration:

sentence=.,10,غ,خميرة,جافة,للخبز,.

ingredient state=[0,0,0,1,2,0,0]

POS tags=[.,B,I,E,F,H,.] coded as [4, 1, 10, 6, 7, 12, 4]

ingredient to be extracted=خميرة جافة

$$\psi_6(i,j) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 3 & 1 & 3 \\ 1 & 0 & 3 & 3 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 3 & 1 & 3 \\ 0 & 0 & 3 & 3 \\ 1 & 3 & 1 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix} \tag{35}$$

$$\psi_6(12,1) = argmax_{1\leq i\leq 4}(\delta_5(12,i) + log(a_{12i1}))$$

$$\begin{aligned}
\delta_5(12,1) + log(a_{1211}) &= max_{1\leq i\leq 4}(\delta_4(12,i) + log(a_{12i1})) + \lambda_{max} log(b_{121}(\text{"للخبز"})) + log(a_{1211}) \\
&= \underbrace{\delta_4(12,2)}_{-139.3} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"}))}_{-3.118} \\
&+ \lambda_{max} \underbrace{log(P(V_r = \text{"للخبز"}/\Gamma_r = 0, T_r = \text{"إسم مجرور"})}_{-36.629} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"إسم مجرور"}))}_{-4.192} \\
&= -183.24
\end{aligned} \tag{36}$$

	.	10	غ	خميرة	جافة	للخبز	.	
case1	-	-	-	-	$\cancel{T_{r-1} = \text{"إسم مجرور"}}$ $\cancel{\Gamma_{r-1} = 1}$	$\Gamma_r = 0$	-	parasite probability
case2	-	-	-	-	-	$T_r = \text{"إسم مجرور"}$ $\Gamma_r = 0$	-	
case3	-	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$	

Table 37: situations table for $\delta_5(12,1) + log(a_{1211})$ for a simple double agent Viterbi without unknown words consideration, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"})$ is a parasite probability because the real POS tag for the word **جافة** is **نعت** and not **إسم مجرور** and the ingredient state of the word **جافة** is $\Gamma_{r-1} = 2$ and not $\Gamma_{r-1} = 1$

$$\begin{aligned}
\delta_5(12,2) + log(a_{1221}) &= max_{1\leq i\leq 4}(\delta_4(12,i) + log(a_{12i2})) + \lambda_{max} log(b_{122}(\text{"للخبز"})) + log(a_{1221}) \\
&= \underbrace{\delta_4(12,4)}_{-138.61} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} = \text{"إسم مجرور"}))}_{-4.077} \\
&+ \lambda_{max} \underbrace{log(P(V_r = \text{"للخبز"}/\Gamma_r = 1, T_r = \text{"إسم مجرور"})}_{-38.016} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"}))}_{-3.118} \\
&= -183.82
\end{aligned} \tag{37}$$

	.	10	غ	خميرة	جافة	للخبز	.	
case1	-	-	-	-	$\cancel{T_{r-1} = \text{"إسم مجرور"}}$ $\cancel{\Gamma_{r-1} = 3}$	$\Gamma_r = 1$	-	parasite probability
case2	-	-	-	-	-	$T_r = \text{"إسم مجرور"}$ $\Gamma_r = 1$	-	
case3	-	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 1$	$\Gamma_r = 0$	

Table 38: situations table for $\delta_5(12,2) + log(a_{1221})$ for a simple double agent Viterbi without unknown words consideration, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 3, T_{r-1} = \text{"إسم مجرور"})$ is a parasite probability because the real POS tag for the word **جافة** is **نعت** and not **إسم مجرور** and the ingredient state of the word **جافة** is $\Gamma_{r-1} = 2$ and not $\Gamma_{r-1} = 3$

Estimating an ingredient state using a double agent Viterbi algorithm with external knowledge transfer :

sentence=.,10,غ,خميرة,جافة,للخبز,.

ingredient state=[0,0,0,1,2,0,0]

POS tags=[.,B,I,E,F,H,.] coded as [4, 1, 10, 6, 7, 12, 4]

ingredient to be extracted=خميرة جافة

$$
\begin{aligned}
\delta_5(12,1) + log(a_{1211}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i1})) + log(b_{121}(\text{"للخبز"})) + log(a_{1211}) \\
&= \underbrace{\delta_4(7,3)}_{-93.42} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-0.851} \\
&+ \underbrace{log(P(V_r = \text{"للخبز"}/\Gamma_r = 0, T_r = \text{"إسم مجرور"}))}_{-36.63} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"إسم مجرور"}))}_{-4.192} \\
&= -135.094
\end{aligned}
\tag{38}
$$

	.	10	غ	خميرة	جافة	للخبز	.
case1	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 2$	$\Gamma_r = 0$	-
case2	-	-	-	-	-	$T_r = \text{"إسم مجرور"}$ $\Gamma_r = 0$	-
case3	-	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$

parasite probability eleminated

Table 39: Situations table for $\delta_5(12,1) + log(a_{1211})$ for a simple double agent Viterbi with external knowledge transfer without unknown words consideration, the parasite probability probability $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"})$ is eliminated and replaced by the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"})$

$$
\begin{aligned}
\delta_5(12,2) + log(a_{1221}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i2})) + log(b_{122}(\text{"للخبز"})) + log(a_{1221}) \\
&= \underbrace{\delta_4(7,3)}_{-93.42} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-6.5} \\
&+ \underbrace{log(P(V_r = \text{"للخبز"}/\Gamma_r = 1, T_r = \text{"إسم مجرور"}))}_{-38.015} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"}))}_{-3.118} \\
&= -141.062
\end{aligned}
\tag{39}
$$

	.	10	غ	خميرة	جافة	للخبز	.
case1	-	-	-	-	$T_{r-1}=$ "نعت" $\Gamma_{r-1}=2$	$\Gamma_r=1$	parasite probability eliminated
case2	-	-	-	-	-	$T_r=$ "إسم مجرور" $\Gamma_r=1$	-
case3	-	-	-	-	-	$T_{r-1}=$ "إسم مجرور" $\Gamma_{r-1}=1$	$\Gamma_r=0$

Table 40: Situations table for $\delta_5(12,2)+log(a_{1221})$ for a simple double agent Viterbi with external knowledge transfer without unknown words consideration, the parasite probability probability $P(\Gamma_r=1/\Gamma_{r-1}=3, T_{r-1}=$ "إسم مجرور"$)$ is eliminated and replaced by the probability $P(\Gamma_r=1/\Gamma_{r-1}=2, T_{r-1}=$ "نعت"$)$

Estimating an ingredient state using a simple double agent Viterbi with unknown words consideration without external knowledge transfer:

sentence=.,10,غ,خميرة,جافة, $\underbrace{\text{للخبز}}_{unknown}$,.

ingredient state=[0,0,0,1,2,0,0]

POS tags=[.,B,I,E,F,H,.] coded as [4,10,11,7,8,9,4]

ingredient to be extracted=خميرة جافة

$$\psi_6(i,j)=\begin{bmatrix}1&0&1&1\\3&0&3&3\\3&3&3&3\\0&0&1&1\\3&0&1&3\\1&0&1&3\\1&3&1&3\\0&0&0&0\\1&3&3&3\\0&0&1&0\\3&0&3&3\\3&3&3&3\\0&3&3&3\\3&3&3&3\end{bmatrix} \tag{40}$$

$$\psi_6(9,1)=argmax_{1\leq i\leq 4}(\delta_5(9,i)+log(a_{9i1}))$$

$$
\begin{aligned}
\delta_5(9,1) + log(a_{911}) &= max_{1\leq i\leq 4}(\delta_4(9,i) + log(a_{9i1})) + \lambda_{max} log(b_{91}(\text{"للخبز"})) + log(a_{911}) \\
&= \underbrace{\delta_4(9,2)}_{-137.679} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"}))}_{-3.155} \\
&+ \underbrace{log(P(V_r = \text{"لل"}/\Gamma_r = 0, T_r = \text{"إسم مجرور"})}_{-6.948} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"إسم مجرور"}))}_{-4.1916} \\
&= -151.97
\end{aligned} \tag{41}
$$

	.	10	غ	خميرة	جافة	للخبز	.
case1	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 1$	$\Gamma_r = 0$	-
case2	-	-	-	-	-	$T_r = \text{"إسم مجرور"}$ $\Gamma_r = 0$	-
case3	-	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 0$	parasite probability $\Gamma_r = 0$

Table 41: Situations table for $\delta_5(9,1) + log(a_{911})$ for a simple double agent Viterbi with unknown words consideration without external knowledge transfer, $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"})$ is a parasite probability because the real ingredient state of the word **جافة** is $\Gamma_{r-1} = 2$ and the real POS tag of the word **جافة** is **نعت**

$$
\begin{aligned}
\delta_5(9,2) + log(a_{921}) &= max_{1\leq i\leq 4}(\delta_4(9,i) + log(a_{9i2})) + \lambda_{max} log(b_{92}(\text{"لل"})) + log(a_{921}) \\
&= \underbrace{\delta_4(9,4)}_{-137.062} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} = \text{"إسم مجرور"}))}_{-4.0604} \\
&+ \underbrace{log(P(V_r = \text{"لل"}/\Gamma_r = 1, T_r = \text{"إسم مجرور"})}_{-6.7473} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"}))}_{-3.155} \\
&= -151.025
\end{aligned} \tag{42}
$$

Estimating an ingredient state using a simple double agent Viterbi algorithm with external knowledge transfer :

sentence=.,10,غ,خميرة,جافة, $\underbrace{\text{للخبز}}_{unknown}$,.

	.	10	غ	خميرة	جافة	للخبز	.
case1	-	-	-	-	$\cancel{T_{r-1} = \text{"إسم مجرور"}}$ $\cancel{\Gamma_{r-1} = 3}$	$\Gamma_r = 1$	-
case2	-	-	-	-	-	$T_r = \text{"إسم مجرور"}$ $\Gamma_r = 1$	-
case3	-	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 1$	$\Gamma_r = 0$

parasite probability

Table 42: Situations table for $\delta_5(9,2) + log(a_{921})$ for a simple double agent Viterbi with unknown words consideration, $P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} = \text{"إسم مجرور"})$ is a parasite probability because the real ingredient state of the word **جافة** is $\Gamma_{r-1} = 2$ and the real POS tag of the word **جافة** is **نعت**

ingredient state=[0,0,0,1,2,0,0]
POS tags=[.,B,I,E,F,H,.] coded as [4,10,11,7,8,9,4]
ingredient to be extracted=**خميرة جافة**

$$\begin{aligned}
\delta_5(9,1) + log(a_{911}) &= max_{1\leq i\leq 4}(\delta_4(POS[5], i) + log(a_{POS[5]i1})) + log(b_{91}(\text{"للخبز"})) + log(a_{911}) \\
&= \underbrace{\delta_4(8,3)}_{-96.916} + \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-0.8461} \\
&+ \underbrace{log(P(V_r = \text{"لل"}/\Gamma_r = 0, T_r = \text{"إسم مجرور"})}_{-6.9488} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} = \text{"إسم مجرور"}))}_{-4.1916} \\
&= -108.902
\end{aligned} \tag{43}$$

	.	10	غ	خميرة	جافة	للخبز	.
case1	-	-	-	-	$T_{r-1} = \text{"نعت"}$ $\Gamma_{r-1} = 2$	$\Gamma_r = 0$	-
case2	-	-	-	-	-	$T_r = \text{"إسم مجرور"}$ $\Gamma_r = 0$	-
case3	-	-	-	-	-	$T_{r-1} = \text{"إسم مجرور"}$ $\Gamma_{r-1} = 0$	$\Gamma_r = 0$

parasite probability eleminated

Table 43: Situations table for $\delta_5(9,1) + log(a_{911})$ for a simple double agent Viterbi with external knowledge transfer with unknown words consideration, the parasite probability probability $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"})$ is eliminated and replaced by the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"})$

$$
\begin{aligned}
\delta_5(9,2) + log(a_{921}) &= max_{1\leq i\leq 4}(\delta_4(POS[5],i) + log(a_{POS[5]i2})) + log(b_{92}(\text{"للخبز"})) + log(a_{921}) \\
&= \underbrace{\delta_4(8,3)}_{-96.91} + \underbrace{log(P(\Gamma_r = 1/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"}))}_{-6.335} \\
&+ \underbrace{log(P(V_r = \text{"لل"}/\Gamma_r = 1, T_r = \text{"إسم مجرور"})}_{-6.747} \\
&+ \underbrace{log(P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} = \text{"إسم مجرور"}))}_{-3.155} \\
&= -113.153
\end{aligned} \tag{44}
$$

	.	10	غ	خميرة	جافة	للخبز	.
case1	-	-	-	-	$T_{r-1} = $ "نعت" $\Gamma_{r-1} = 2$	$\Gamma_r = 1$	parasite probability eliminated
case2	-	-	-	-	-	$T_r = $ "إسم مجرور" $\Gamma_r = 1$	-
case3	-	-	-	-	-	$T_{r-1} = $ "إسم مجرور" $\Gamma_{r-1} = 1$	$\Gamma_r = 0$

Table 44: Situations table for $\delta_5(9,2) + log(a_{921})$ for a primitive double agent Viterbi with external knowledge transfer with unknown words consideration, the parasite probability probability $P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} = \text{"إسم مجرور"})$ is eliminated and replaced by the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 2, T_{r-1} = \text{"نعت"})$

4 Results and Discussion

When comparing the results obtained in table Table 31, we found that the double layered Viterbi algorithm with external knowledge transfer is overcoming the 3 other methods in terms of precision with a f-score value of 96,32 % versus 70,06 % for a first order mono-layered HMM model, 68,18 % for a second order mono-layered HMM model and 81,32 % for a simple double agent Viterbi algorithm. This can be explained by the external knowledge transfer characterizing our developed algorithm that minimized the amount of parasite probabilities that may appear during the calculation of the most probable elements of a state matrix. For example for the sentence "للخبز جافة خميرة غ10" where all the words are known both a simple double agent Viterbi and a simple double agent Viterbi with external knowlage transfer estimates well the token للخبز which is not an ingredient, but for the first algorithm we have $|X - Y| = |\delta_5(12,1) + log(a_{1211}) - \delta_5(12,2) - log(a_{1221})| = 0,58$ and we have for the second one $|X - Y| = |\delta_5(12,1) + log(a_{1211}) - \delta_5(12,2) - log(a_{1221})| = 5,97$ (figure 3) which means that the state matrix for a double agent Viterbi algorithm with

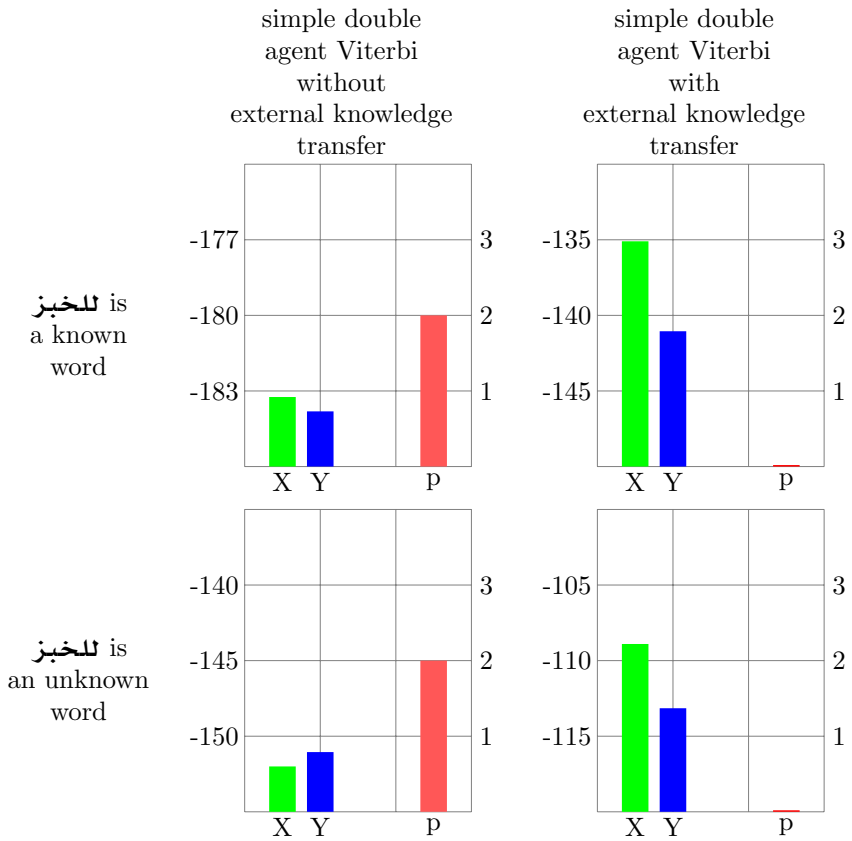


Figure 3: data sensitivity bar plot for the word للخبز inside the sentence ٠١ غ خميرة جافة للخبز under graduated conditions. X is the first most probable case inside a state matrix (للخبز is not an ingredient), Y is the second most probable case inside a state matrix (للخبز is an ingredient constituted with a single word), P is the number of parasite probabilities observed when calculating X and Y.

external knowledge transfer differentiate well between the most probable cases inside a state matrix and it is more confident in estimating an ingredient state. We can explain the efficiency of our new algorithm by the fact that when the probabilities constituting the equation of the most probable elements of the state matrix are describing well the values hold by a token and not wrong parameters like we see in a parasite probability, the elements of the state matrix tend to enlarge the differences between a state describing the real estimation and a state describing a wrong estimation.

One important characteristic found in a double agent Viterbi algorithm with external knowledge transfer is its capacity to well estimate ingredients state against a mono-layered Viterbi algorithm and a simple double agent Viterbi algorithm without external knowledge transfer in a context of unknown words. When an unknown word is estimated, the lexical probability $P(V_r = $"للخبز"$/\Gamma_r = 0, T_r = $"إسم مجرور"$)$ becomes $P(V_r = $"لل"$/\Gamma_r = 0, T_r = $"إسم مجرور"$)$ and its importance for estimating hidden states on the second layer of our model becomes lower . External knowledge transfer allowed us to clean our double layered Viterbi algorithm from parasite probabilities and maintain a good f-score of 72.19 % (table 36) because we are confirming that knowledge transfer is increasing the precision comparing to what we got for our ingredient extractor (Baklouti, 2021)(table 36). Best accuracy for unknown words is detained by our new algorithm (table 36) . These promising performances can be explained by observing the equations determining the most probable cases of a state matrix. For example , in the sentence "غ خميرة جافة للخبز10" when the token للخبز is unknown , a primitive Viterbi algorithm does not estimated the exact ingredient state for that token which is $\Gamma = 0$ given that the word للخبز is not an ingredient , on the other side , a double layered Viterbi algorithm with internal knowledge transfer estimated well the ingredient state , this can be explained by the absence of parasite probabilities in the second algorithm given that the lexical probabilities $P(V_r = $"لل"$/\Gamma_r = 0, T_r = $"إسم مجرور"$)$ and $P(V_r = $"لل"$/\Gamma_r = 1, T_r = $"إسم مجرور"$)$ keep the same values when they are used in the a double agent Viterbi algorithm with and without external knowledge transfer.

Another concept characterizing a simple double agent Viterbi algorithm with external knowledge transfer is its capacity to react effectively toward dataset tuning and its capacity to resist data monotonicity logic. For example, our dataset is describing the ingredients used in a recipe, and to achieve a better understanding, a typical sentence structure is frequently used to present an ingredient like using the preposition "of" and then announcing the wanted information but sometimes this monotonous logic is not always true when the preposition "of" is used to describe the ingredient like in the sentence "olive oil of excellent quality". We created an adapted graph to study the comportment of our new algorithm toward monotonous logic called a data sensitivity bar plot. A data sensitivity bar plot is presenting two pieces of information: the value of the most probable cases inside a state matrix and the number of parasite probabilities observed. That information is studied under graduated conditions. A graduate condition is used to explore the dynamical comport-

ment of the algorithm, for example, we have in our dataset the sentence "olive oil of excellent quality" where the ingredient is situated at the beginning of the sentence and it is hard for any algorithm to give a correct prediction for this sentence when it is tagged only one time on the tag-set especially for a mono-agent Viterbi algorithm where the only parameters of this model are the tokens and the ingredients states. Data tuning is one of the graduated conditions we can use by adding first a similar sentence that keeps the same tags for part of speech and for ingredients state like the sentence "one tomato of generous size", and second by adding the same sentence to obtain a tag-set with the sentence "olive oil of excellent quality" repeated twice. We can associate this graduated condition with the algorithm used to predict ingredients states and deployed in a data sensitivity plot gradually from the method with the lowest f-score to the algorithm with the highest f-score (figure 2). Data sensitivity bar plot with the previously described combination shows a certain static behavior for a second-order mono-agent Viterbi and a simple double agent Viterbi, without external knowledge transfer. Dynamic behavior is observed only for a simple double agent Viterbi without external knowledge transfer where the value of the correct state increase and the number of parasite probabilities decrease when we move from a condition with low grade to a condition with high grade. Moving from known to unknown words could be also a graduated condition used to analyze the dynamic behavior of an algorithm in a data sensitivity bar plot (figure 3).

5 Conclusion

We have proposed an algorithm mixing Hidden Markov Models and external knowledge transfer to solve the problem of information extraction in natural language processing, our algorithm can be used in other problems where Hidden Markov Models are deployed. This algorithm has proven a spectacular improvement in f1 score without unknown words consideration and performances for unknown words are improved comparing to old results for simple double agents Viterbi without external knowledge transfer (Baklouti, 2021) but still can be improved by creating a full double agent Viterbi with external knowledge transfer where the number of parasite probabilities generated is less than we obtained on this paper.

A Detailed Iterations of a simple double agent Viterbi Algorithm with external knowledge transfer

1. Predicting POS Tags for the first Layer:

   We can use for predicting POS tags any method that provide a high accuracy and f-score. Some features used for natural language programming problems are directly extracted from the tokens like extracting the first letters from a token.

2. Predicting Ingredients state for the second Layer:

   The variables:

   - $\delta_l(i,j) = max_{\tau_1...\tau_{l-1}} P(\tau_1...\tau_l = [t_i, \gamma_j]/v_1...v_l, t_1...t_l, y_1...y_l), l = 1...L$
   - $\psi_l(i,j) = argmax_{\tau_1...\tau_{l-1}} P(\tau_1...\tau_l = [t_i, \gamma_j]/v_1...v_l, t_1...t_l, y_1...y_l), l = 1...L$

   The procedure:

   (1) Initialization step:

$$\delta_1(i,j) = \begin{cases} \pi_j b_{ij}(v_1) & v_1 \text{ is known i=1..14 j=1..4} \\ \\ \pi_j c_{ij}(v_1) & v_1 \text{ is unknown i=1..14 j=1..4} \end{cases}$$

$$\psi_1(i,j) = 0 \text{ i=1..14 j=1..4}$$

   (2) Iteration step:

$$\delta_l(i,j) = \begin{cases} max_j[\delta_{l-1}(y(l),j)a_{y(l)jk}]b_{ij}(v_l) & v_l \text{ is known , l=2...L} \\ & \text{j=1...4, i=1..14} \\ & \text{k=1...4, y(l) is the tag} \\ & \text{calculated at previous layer} \\ & \text{in position l} \\ \\ max_j[\delta_{l-1}(y(l),j)a_{y(l)jk}]c_{ij}(v_l) & v_l \text{ is unknown , l=2...L} \\ & \text{j=1...4, i=1..14} \\ & \text{k=1...4, y(l) is the tag} \\ & \text{calculated at previous layer} \\ & \text{in position l} \end{cases}$$

$$\psi_l(i,j) = argmax_j[\delta_{l-1}(y(l),j)a_{y(l)jk}]$$

j=1..4,i=1..14,k=1..4,y(l) is the tag calculated at previous layer in position l

(3) Termination:

$$\tau_L^* = argmax_{i=y(L),j=1..4}\delta_L(i,j)$$

where y(L) is the tag calculated at previous layer in position L

(4) Backtracking:

$$\tau_l^* = \psi_l(y(l),\tau_{l+1}^*) \text{ l=L-1..2,1}$$

where y(l) is the tag calculated at previous layer in position l

3. Predicting Ingredients state for the second Layer in Log-space and introduction of the hyper-parameter λ

The variables:

- $\delta_l(i,j) = max_{\tau_1...\tau_{l-1}}P(\tau_1...\tau_l = [t_i,\gamma_j]/v_1...v_l,t_1...t_l,y_1...y_l), l = 1...L$
- $\psi_l(i,j) = argmax_{\tau_1...\tau_{l-1}}P(\tau_1...\tau_l = [t_i,\gamma_j]/v_1...v_l,t_1...t_l,y_1...y_l), l = 1...L$

The procedure:

(1) Initialization step:

$$\delta_1(i,j) = \begin{cases} \log(\pi_j) + \log(b_{ij}(v_1)) & v_1 \text{ is known i=1..14 j=1..4} \\ \\ \log(\pi_j) + \log(c_{ij}(v_1)) & v_1 \text{ is unknown i=1..14 j=1..4} \end{cases}$$

$$\psi_1(i,j) = 0 \text{ i=1..14 j=1..4}$$

(2) Iteration step:

$$\delta_l(i,j) = \begin{cases} max_j[\delta_{l-1}(y(l),j) + \log(a_{y(l)jk})] + \lambda_{max}\log(b_{ij}(v_l)) & v_l \text{ is known l=2..L} \\ & \text{j=1...4, i=1..14} \\ & \text{k=1...4, y(l) is the tag} \\ & \text{calculated at previous layer} \\ & \text{in position l} \\ \\ max_j[\delta_{l-1}(y(l),j) + \log(a_{y(l)jk})] + \lambda_{max}\log(c_{ij}(v_l)) & v_l \text{ is unknown} \\ & \text{l=2...L j=1..4} \\ & \text{i=1..14 k=1..4, y(l) is the tag} \\ & \text{calculated at previous layer} \\ & \text{in position l} \end{cases}$$

$$\psi_l(i,j) = argmax_j[\delta_{l-1}(y(l),j) + \log(a_{y(l)jk})] \text{ j=1..4,i=1..14,k=1..4,y(l) is the tag calculated at previous layer in position l}$$

(3) Termination:

$$\tau_L^* = argmax_{i=y(L),j=1..4}\delta_L(i,j)$$

where y(L) is the tag calculated at previous layer in position L

(4) Backtracking:

$$\tau_l^* = \psi_l(y(l),\tau_{l+1}^*) \text{ l=L-1..2,1}$$

where y(l) is the tag calculated at previous layer in position l

- Initialization step:

$$\begin{bmatrix} d_{1,1,1} & d_{1,1,2} & d_{1,1,3} & d_{1,1,4} \\ d_{1,2,1} & d_{1,2,2} & d_{1,2,3} & d_{1,2,4} \\ d_{1,3,1} & d_{1,3,2} & d_{1,3,3} & d_{1,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1,14,1} & d_{1,14,2} & d_{1,14,3} & d_{1,14,4} \end{bmatrix} \begin{bmatrix} pi_1 & pi_2 & pi_3 & pi_4 \end{bmatrix} \begin{bmatrix} b_{1,1,x[1]} & b_{1,2,x[1]} & b_{1,3,x[1]} & b_{1,4,x[1]} \\ b_{2,1,x[1]} & b_{2,2,x[1]} & b_{2,3,x[1]} & b_{2,4,x[1]} \\ b_{3,1,x[1]} & b_{3,2,x[1]} & b_{3,3,x[1]} & b_{3,4,x[1]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[1]} & b_{14,2,x[1]} & b_{14,3,x[1]} & b_{14,4,x[1]} \end{bmatrix}$$

$$d_{1,1,1} = pi_1 \times b_{1,1,x[1]}$$

$$\begin{bmatrix} d_{1,1,1} & d_{1,1,2} & d_{1,1,3} & d_{1,1,4} \\ d_{1,2,1} & d_{1,2,2} & d_{1,2,3} & d_{1,2,4} \\ d_{1,3,1} & d_{1,3,2} & d_{1,3,3} & d_{1,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1,14,1} & d_{1,14,2} & d_{1,14,3} & d_{1,14,4} \end{bmatrix} \begin{bmatrix} pi_1 & pi_2 & pi_3 & pi_4 \end{bmatrix} \begin{bmatrix} b_{1,1,x[1]} & b_{1,2,x[1]} & b_{1,3,x[1]} & b_{1,4,x[1]} \\ b_{2,1,x[1]} & b_{2,2,x[1]} & b_{2,3,x[1]} & b_{2,4,x[1]} \\ b_{3,1,x[1]} & b_{3,2,x[1]} & b_{3,3,x[1]} & b_{3,4,x[1]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[1]} & b_{14,2,x[1]} & b_{14,3,x[1]} & b_{14,4,x[1]} \end{bmatrix}$$

$$d_{1,1,2} = pi_2 \times b_{1,2,x[1]}$$

$$\begin{bmatrix} d_{1,1,1} & d_{1,1,2} & d_{1,1,3} & d_{1,1,4} \\ d_{1,2,1} & d_{1,2,2} & d_{1,2,3} & d_{1,2,4} \\ d_{1,3,1} & d_{1,3,2} & d_{1,3,3} & d_{1,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1,14,1} & d_{1,14,2} & d_{1,14,3} & d_{1,14,4} \end{bmatrix} \begin{bmatrix} pi_1 & pi_2 & pi_3 & pi_4 \end{bmatrix} \begin{bmatrix} b_{1,1,x[1]} & b_{1,2,x[1]} & b_{1,3,x[1]} & b_{1,4,x[1]} \\ b_{2,1,x[1]} & b_{2,2,x[1]} & b_{2,3,x[1]} & b_{2,4,x[1]} \\ b_{3,1,x[1]} & b_{3,2,x[1]} & b_{3,3,x[1]} & b_{3,4,x[1]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[1]} & b_{14,2,x[1]} & b_{14,3,x[1]} & b_{14,4,x[1]} \end{bmatrix}$$

$$d_{1,1,3} = pi_3 \times b_{1,3,x[1]}$$

$$\begin{bmatrix} d_{1,1,1} & d_{1,1,2} & d_{1,1,3} & d_{1,1,4} \\ d_{1,2,1} & d_{1,2,2} & d_{1,2,3} & d_{1,2,4} \\ d_{1,3,1} & d_{1,3,2} & d_{1,3,3} & d_{1,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1,14,1} & d_{1,14,2} & d_{1,14,3} & d_{1,14,4} \end{bmatrix} \begin{bmatrix} pi_1 & pi_2 & pi_3 & pi_4 \end{bmatrix} \begin{bmatrix} b_{1,1,x[1]} & b_{1,2,x[1]} & b_{1,3,x[1]} & b_{1,4,x[1]} \\ b_{2,1,x[1]} & b_{2,2,x[1]} & b_{2,3,x[1]} & b_{2,4,x[1]} \\ b_{3,1,x[1]} & b_{3,2,x[1]} & b_{3,3,x[1]} & b_{3,4,x[1]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[1]} & b_{14,2,x[1]} & b_{14,3,x[1]} & b_{14,4,x[1]} \end{bmatrix}$$

$$d_{1,1,4} = pi_4 \times b_{1,4,x[1]}$$

$$\begin{bmatrix} d_{1,1,1} & d_{1,1,2} & d_{1,1,3} & d_{1,1,4} \\ d_{1,2,1} & d_{1,2,2} & d_{1,2,3} & d_{1,2,4} \\ d_{1,3,1} & d_{1,3,2} & d_{1,3,3} & d_{1,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1,14,1} & d_{1,14,2} & d_{1,14,3} & d_{1,14,4} \end{bmatrix} \begin{bmatrix} pi_1 & pi_2 & pi_3 & pi_4 \end{bmatrix} \begin{bmatrix} b_{1,1,x[1]} & b_{1,2,x[1]} & b_{1,3,x[1]} & b_{1,4,x[1]} \\ b_{2,1,x[1]} & b_{2,2,x[1]} & b_{2,3,x[1]} & b_{2,4,x[1]} \\ b_{3,1,x[1]} & b_{3,2,x[1]} & b_{3,3,x[1]} & b_{3,4,x[1]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[1]} & b_{14,2,x[1]} & b_{14,3,x[1]} & b_{14,4,x[1]} \end{bmatrix}$$

$$d_{1,2,1} = pi_1 \times b_{2,1,x[1]}$$

$$\vdots$$

$$\begin{bmatrix} d_{1,1,1} & d_{1,1,2} & d_{1,1,3} & d_{1,1,4} \\ d_{1,2,1} & d_{1,2,2} & d_{1,2,3} & d_{1,2,4} \\ d_{1,3,1} & d_{1,3,2} & d_{1,3,3} & d_{1,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{1,14,1} & d_{1,14,2} & d_{1,14,3} & d_{1,14,4} \end{bmatrix} \begin{bmatrix} pi_1 & pi_2 & pi_3 & pi_4 \end{bmatrix} \begin{bmatrix} b_{1,1,x[1]} & b_{1,2,x[1]} & b_{1,3,x[1]} & b_{1,4,x[1]} \\ b_{2,1,x[1]} & b_{2,2,x[1]} & b_{2,3,x[1]} & b_{2,4,x[1]} \\ b_{3,1,x[1]} & b_{3,2,x[1]} & b_{3,3,x[1]} & b_{3,4,x[1]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[1]} & b_{14,2,x[1]} & b_{14,3,x[1]} & b_{14,4,x[1]} \end{bmatrix}$$

$$d_{1,14,4} = pi_4 \times b_{14,4,x[1]}$$

- Iteration step:

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,1,1} = max(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1}) \times b_{1,1,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right)$$

$$p_{2,1,1} = argmax(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1})$$

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,2,1} = max(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1}) \times b_{2,1,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right)$$

$$p_{2,2,1} = argmax(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1})$$

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,3,1} = max(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1}) \times b_{3,1,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right)$$

$$p_{2,3,1} = argmax(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1})$$

$$\vdots$$

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,14,1} = max(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1}) \times b_{14,1,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,1} a_{y[2],2,1} a_{y[2],3,1} a_{y[2],4,1} \end{bmatrix} \right)$$

$$p_{2,14,1} = argmax(d_{1,y[2],1} \times a_{y[2],1,1}, d_{1,y[2],2} \times a_{y[2],2,1}, \cdots, d_{1,y[2],4} \times a_{y[2],4,1})$$

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,2} a_{y[2],2,2} a_{y[2],3,2} a_{y[2],4,2} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,1,2} = max(d_{1,y[2],1} \times a_{y[2],1,2}, d_{1,y[2],2} \times a_{y[2],2,2}, \cdots, d_{1,y[2],4} \times a_{y[2],4,2}) \times b_{1,2,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,2} a_{y[2],2,2} a_{y[2],3,2} a_{y[2],4,2} \end{bmatrix} \right)$$

$$p_{2,1,2} = argmax(d_{1,y[2],1} \times a_{y[2],1,2}, d_{1,y[2],2} \times a_{y[2],2,2}, \cdots, d_{1,y[2],4} \times a_{y[2],4,2})$$

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,2} a_{y[2],2,2} a_{y[2],3,2} a_{y[2],4,2} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,2,2} = max(d_{1,y[2],1} \times a_{y[2],1,2}, d_{1,y[2],2} \times a_{y[2],2,2}, \cdots, d_{1,y[2],4} \times a_{y[2],4,2}) \times b_{2,2,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,2} a_{y[2],2,2} a_{y[2],3,2} a_{y[2],4,2} \end{bmatrix} \right)$$

$$p_{2,2,2} = argmax(d_{1,y[2],1} \times a_{y[2],1,2}, d_{1,y[2],2} \times a_{y[2],2,2}, \cdots, d_{1,y[2],4} \times a_{y[2],4,2})$$

$$\vdots$$

$$\begin{bmatrix} d_{2,1,1} & d_{2,1,2} & d_{2,1,3} & d_{2,1,4} \\ d_{2,2,1} & d_{2,2,2} & d_{2,2,3} & d_{2,2,4} \\ d_{2,3,1} & d_{2,3,2} & d_{2,3,3} & d_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{2,14,1} & d_{2,14,2} & d_{2,14,3} & d_{2,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,4} a_{y[2],2,4} a_{y[2],3,4} a_{y[2],4,4} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[2]} & b_{1,2,x[2]} & b_{1,3,x[2]} & b_{1,4,x[2]} \\ b_{2,1,x[2]} & b_{2,2,x[2]} & b_{2,3,x[2]} & b_{2,4,x[2]} \\ b_{3,1,x[2]} & b_{3,2,x[2]} & b_{3,3,x[2]} & b_{3,4,x[2]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[2]} & b_{14,2,x[2]} & b_{14,3,x[2]} & b_{14,4,x[2]} \end{bmatrix}$$

$$d_{2,14,4} = max(d_{1,y[2],1} \times a_{y[2],1,4}, d_{1,y[2],2} \times a_{y[2],2,4}, \cdots, d_{1,y[2],4} \times a_{y[2],4,4}) \times b_{14,4,x[2]}$$

$$\begin{bmatrix} p_{2,1,1} & p_{2,1,2} & p_{2,1,3} & p_{2,1,4} \\ p_{2,2,1} & p_{2,2,2} & p_{2,2,3} & p_{2,2,4} \\ p_{2,3,1} & p_{2,3,2} & p_{2,3,3} & p_{2,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{2,14,1} & p_{2,14,2} & p_{2,14,3} & p_{2,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{1,y[2],1} d_{1,y[2],2} d_{1,y[2],3} d_{1,y[2],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[2],1,4} a_{y[2],2,4} a_{y[2],3,4} a_{y[2],4,4} \end{bmatrix} \right)$$

$$p_{2,14,4} = argmax(d_{1,y[2],1} \times a_{y[2],1,4}, d_{1,y[2],2} \times a_{y[2],2,4}, \cdots, d_{1,y[2],4} \times a_{y[2],4,4})$$

$$\begin{bmatrix} d_{3,1,1} & d_{3,1,2} & d_{3,1,3} & d_{3,1,4} \\ d_{3,2,1} & d_{3,2,2} & d_{3,2,3} & d_{3,2,4} \\ d_{3,3,1} & d_{3,3,2} & d_{3,3,3} & d_{3,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{3,14,1} & d_{3,14,2} & d_{3,14,3} & d_{3,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{2,y[3],1} & d_{2,y[3],2} & d_{2,y[3],3} & d_{2,y[3],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[3],1,1} & a_{y[3],2,1} & a_{y[3],3,1} & a_{y[3],4,1} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[3]} & b_{1,2,x[3]} & b_{1,3,x[3]} & b_{1,4,x[3]} \\ b_{2,1,x[3]} & b_{2,2,x[3]} & b_{2,3,x[3]} & b_{2,4,x[3]} \\ b_{3,1,x[3]} & b_{3,2,x[3]} & b_{3,3,x[3]} & b_{3,4,x[3]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[3]} & b_{14,2,x[3]} & b_{14,3,x[3]} & b_{14,4,x[3]} \end{bmatrix}$$

$$d_{3,1,1} = max(d_{2,y[3],1} \times a_{y[3],1,1}, d_{2,y[3],2} \times a_{y[3],2,1}, \cdots, d_{2,y[3],4} \times a_{y[3],4,1}) \times b_{1,1,x[3]}$$

$$\begin{bmatrix} p_{3,1,1} & p_{3,1,2} & p_{3,1,3} & p_{3,1,4} \\ p_{3,2,1} & p_{3,2,2} & p_{3,2,3} & p_{3,2,4} \\ p_{3,3,1} & p_{3,3,2} & p_{3,3,3} & p_{3,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{3,14,1} & p_{3,14,2} & p_{3,14,3} & p_{3,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{2,y[3],1} & d_{2,y[3],2} & d_{2,y[3],3} & d_{2,y[3],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[3],1,1} & a_{y[3],2,1} & a_{y[3],3,1} & a_{y[3],4,1} \end{bmatrix} \right)$$

$$p_{3,1,1} = argmax(d_{2,y[3],1} \times a_{y[3],1,1}, d_{2,y[3],2} \times a_{y[3],2,1}, \cdots, d_{2,y[3],4} \times a_{y[3],4,1})$$

$$\vdots$$

$$\begin{bmatrix} d_{L,1,1} & d_{L,1,2} & d_{L,1,3} & d_{L,1,4} \\ d_{L,2,1} & d_{L,2,2} & d_{L,2,3} & d_{L,2,4} \\ d_{L,3,1} & d_{L,3,2} & d_{L,3,3} & d_{L,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{L,14,1} & d_{L,14,2} & d_{L,14,3} & d_{L,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{L-1,y[L],1} & d_{L-1,y[L],2} & d_{L-1,y[L],3} & d_{L-1,y[L],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[L],1,1} & a_{y[L],2,1} & a_{y[L],3,1} & a_{y[L],4,1} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[L]} & b_{1,2,x[L]} & b_{1,3,x[L]} & b_{1,4,x[L]} \\ b_{2,1,x[L]} & b_{2,2,x[L]} & b_{2,3,x[L]} & b_{2,4,x[L]} \\ b_{3,1,x[L]} & b_{3,2,x[L]} & b_{3,3,x[L]} & b_{3,4,x[L]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[L]} & b_{14,2,x[L]} & b_{14,3,x[L]} & b_{14,4,x[L]} \end{bmatrix}$$

$$d_{L,1,1} = max(d_{L-1,y[L],1} \times a_{y[L],1,1}, d_{L-1,y[L],2} \times a_{y[L],2,1}, \cdots, d_{L-1,y[L],4} \times a_{y[L],4,1}) \times b_{1,1,x[L]}$$

$$\begin{bmatrix} p_{L,1,1} & p_{L,1,2} & p_{L,1,3} & p_{L,1,4} \\ p_{L,2,1} & p_{L,2,2} & p_{L,2,3} & p_{L,2,4} \\ p_{L,3,1} & p_{L,3,2} & p_{L,3,3} & p_{L,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{L,14,1} & p_{L,14,2} & p_{L,14,3} & p_{L,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{L-1,y[L],1} & d_{L-1,y[L],2} & d_{L-1,y[L],3} & d_{L-1,y[L],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[L],1,1} & a_{y[L],2,1} & a_{y[L],3,1} & a_{y[L],4,1} \end{bmatrix} \right)$$

$$p_{L,1,1} = argmax(d_{L-1,y[L],1} \times a_{y[L],1,1}, d_{L-1,y[L],2} \times a_{y[L],2,1}, \cdots, d_{L-1,y[L],4} \times a_{y[L],4,1})$$

$$\vdots$$

$$\begin{bmatrix} d_{L,1,1} & d_{L,1,2} & d_{L,1,3} & d_{L,1,4} \\ d_{L,2,1} & d_{L,2,2} & d_{L,2,3} & d_{L,2,4} \\ d_{L,3,1} & d_{L,3,2} & d_{L,3,3} & d_{L,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{L,14,1} & d_{L,14,2} & d_{L,14,3} & d_{L,14,4} \end{bmatrix} = max_i \left(\begin{bmatrix} d_{L-1,y[L],1} & d_{L-1,y[L],2} & d_{L-1,y[L],3} & d_{L-1,y[L],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[L],1,4} & a_{y[L],2,4} & a_{y[L],3,4} & a_{y[L],4,4} \end{bmatrix} \right) \cdot \begin{bmatrix} b_{1,1,x[L]} & b_{1,2,x[L]} & b_{1,3,x[L]} & b_{1,4,x[L]} \\ b_{2,1,x[L]} & b_{2,2,x[L]} & b_{2,3,x[L]} & b_{2,4,x[L]} \\ b_{3,1,x[L]} & b_{3,2,x[L]} & b_{3,3,x[L]} & b_{3,4,x[L]} \\ \vdots & \vdots & \vdots & \vdots \\ b_{14,1,x[L]} & b_{14,2,x[L]} & b_{14,3,x[L]} & b_{14,4,x[L]} \end{bmatrix}$$

$$d_{L,14,4} = max(d_{L-1,y[L],1} \times a_{y[L],1,4}, d_{L-1,y[L],2} \times a_{y[L],2,4}, \cdots, d_{L-1,y[L],4} \times a_{y[L],4,4}) \times b_{14,4,x[L]}$$

$$\begin{bmatrix} p_{L,1,1} & p_{L,1,2} & p_{L,1,3} & p_{L,1,4} \\ p_{L,2,1} & p_{L,2,2} & p_{L,2,3} & p_{L,2,4} \\ p_{L,3,1} & p_{L,3,2} & p_{L,3,3} & p_{L,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ p_{L,14,1} & p_{L,14,2} & p_{L,14,3} & p_{L,14,4} \end{bmatrix} = argmax_i \left(\begin{bmatrix} d_{L-1,y[L],1} & d_{L-1,y[L],2} & d_{L-1,y[L],3} & d_{L-1,y[L],4} \end{bmatrix} \cdot \begin{bmatrix} a_{y[L],1,1} & a_{y[L],2,1} & a_{y[L],3,1} & a_{y[L],4,1} \end{bmatrix} \right)$$

$$p_{L,14,4} = argmax(d_{L-1,y[L],1} \times a_{y[L],1,4}, d_{L-1,y[L],2} \times a_{y[L],2,4}, \cdots, d_{L-1,y[L],4} \times a_{y[L],4,4})$$

- Termination:

$$\begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_{L-2} & \tau_{L-1} & \tau_L \end{bmatrix} = arg_j max_{1 \le i \le 14, 1 \le j \le 4} \left(\begin{bmatrix} d_{L,1,1} & d_{L,1,2} & d_{L,1,3} & d_{L,1,4} \\ d_{L,2,1} & d_{L,2,2} & d_{L,2,3} & d_{L,2,4} \\ d_{L,3,1} & d_{L,3,2} & d_{L,3,3} & d_{L,3,4} \\ \vdots & \vdots & \vdots & \vdots \\ d_{L,14,1} & d_{L,14,2} & d_{L,14,3} & d_{L,14,4} \end{bmatrix} \right)$$

- Backtracking:

$$\begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_{L-2} & \tau_{L-1} & \tau_L \end{bmatrix} = \psi(L, Y_L, \tau_L)$$

$$\vdots$$

$$\begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_{L-2} & \tau_{L-1} & \tau_L \end{bmatrix} = \psi(2, Y_2, \tau_2)$$

List of Figures

1 data sensitivity bar plot for the word **الجيدة** inside the sentence **زيت زيتون من النوعية الجيدة** under graduated conditions. X is the first most probable case inside a state matrix **الجيدة** is not an ingredient), Y is the second most probable case inside a state matrix **الجيدة** is an ingredient constituted with a single word) and Z is the third most probable case inside a state matrix (**الجيدة** is the second word of an ingredient constituted with two elements) , P is the number of parasite probabilities observed when calculating X and Y and Z. 26

2 Performance comparison between a double agent Viterbi algorithm with external knowledge transfer and other mono layered Viterbi algorithm 30

3 data sensitivity bar plot for the word **للخبز** inside the sentence **٠١ غ خميرة جافة للخبز** under graduated conditions. X is the first most probable case inside a state matrix (**للخبز** is not an ingredient), Y is the second most probable case inside a state matrix (**للخبز** is an ingredient constituted with a single word), P is the number of parasite probabilities observed when calculating X and Y. 41

List of Tables

1 situations table for $\delta_6(1,1)$ for a second order HMM estimation with all the words are known . The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)$ are describing existing cases from the tagged sentence, they are not parasite probabilities. . . 4

2 Situations table for $\delta_6(1,2)$ for a second order HMM estimation. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)$ are describing existing cases from the tagged sentence, supposing that the word **الجيدة** is an ingredient . These probabilities are not parasite probabilities. 5

3 Situations table for $\delta_6(3,1)$ for a second order HMM estimation. The probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, \Gamma_{r-2} = 1)$ is a parasite probability because $\Gamma_{r-2} = 1$ is not the real state of the word **النوعية** and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)$ is not a parasite probability because it is describing an existing case from the tagged sentence after considering the word **الجيدة** as the second word of an ingredient. 5

4 situations table for $\delta_6(1,1)$ for a second order HMM estimation with all the words are known after adding the sentence زيت زيتون من النوعية الجيدة to the dataset. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)$ are describing existing cases from the tagged sentence, they are not parasite probabilities. 6
5 Situations table for $\delta_6(2,1)$ for a second order HMM estimation with all the words are known after adding the sentence زيت زيتون من النوعية الجيدة to the dataset. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)$ are describing existing cases from the tagged sentence, supposing that the word الجيدة is an ingredient . These probabilities are not parasite probabilities. 7
6 Situations table for $\delta_6(3,1)$ for a second order HMM estimation with all the words are known after adding the sentence زيت زيتون من النوعية الجيدة to the dataset. The probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, \Gamma_{r-2} = 1)$ is a parasite probability because $\Gamma_{r-2} = 1$ is not the real state of the word النوعية and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)$ is not a parasite probability because it is describing an existing case from the tagged sentence after considering the word الجيدة as the second word of an ingredient. 7
7 situations table for $\delta_6(1,1)$ for a second order HMM estimation with all the words are known after adding the sentence كعبة طماطم من الحجم الكبير to the dataset. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 0, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 0)$ are describing existing cases from the tagged sentence, they are not parasite probabilities. 8
8 Situations table for $\delta_6(2,1)$ for a second order HMM estimation with all the words are known after adding the sentence كعبة طماطم من الحجم الكبير to the dataset. The probabilities $P(\Gamma_r = 0/\Gamma_{r-1} = 1, \Gamma_{r-2} = 0)$ and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 1)$ are describing existing cases from the tagged sentence, supposing that the word الجيدة is an ingredient . These probabilities are not parasite probabilities. 9
9 Situations table for $\delta_6(3,1)$ for a second order HMM estimation with all the words are known after adding the sentence كعبة طماطم من الحجم الكبير to the dataset. The probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, \Gamma_{r-2} = 1)$ is a parasite probability because $\Gamma_{r-2} = 1$ is not the real state of the word النوعية and $P(V_r = "."/\Gamma_r = 0, \Gamma_{r-1} = 2)$ is not a parasite probability because it is describing an existing case from the tagged sentence after considering the word الجيدة as the second word of an ingredient. 9

10 situations table for $\delta_5(7,1) + log(a_{711})$ for a second order HMM estimation, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 12
11 situations table for $\delta_5(7,2) + log(a_{721})$ for a second order HMM estimation, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 12
12 situations table for $\delta_5(7,3) + log(a_{731})$ for a second order HMM estimation, the probability $P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 13
13 situations table for $\delta_5(7,1)+log(a_{711})$ for a second order HMM estimation after adding the sentence زيت زيتون من النوعية الجيدة to the dataset , the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 15
14 situations table for $\delta_5(7,2)+log(a_{721})$ for a second order HMM estimation after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 15
15 situations table for $\delta_5(7,3)+log(a_{731})$ for a second order HMM estimation after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, the probability $P(\Gamma_r = 2/\Gamma_{r-1} = 3, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت and the ingredient state of the word النوعية is $\Gamma_r = 0$ and not $\Gamma_r = 3$ 16
16 situations table for $\delta_5(7,1)+log(a_{711})$ for a second order HMM estimation after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 18
17 situations table for $\delta_5(7,2)+log(a_{721})$ for a second order HMM estimation after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 18
18 situations table for $\delta_5(7,3)+log(a_{731})$ for a second order HMM estimation after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 0, T_{r-1} =$ "نعت") is a parasite probability because the POS tag of the word النوعية is منعوت and not نعت 19

19 situations table for $\delta_5(7,1)+log(a_{711})$ for a double agent Viterbi with external knowledge transfer, $P(\Gamma_r=0/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت") is a parasite probability because the ingredient state of the word النوعية is $\Gamma_{r-1}=0$ and not $\Gamma_{r-1}=1$ 20
20 situations table for $\delta_5(7,2)+log(a_{721})$ for a double agent Viterbi with external knowledge transfer, $P(\Gamma_r=1/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت") is a sain probability because the word النوعية is a منعوت and it is not an ingredient. 21
21 situations table for $\delta_5(7,3)+log(a_{731})$ for a double agent Viterbi with external knowledge transfer, $P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت") is a parasite probability because the word النوعية is not an ingredient. 21
22 situations table for $\delta_5(7,1)+log(a_{711})$ for a double agent Viterbi with external knowledge transfer, after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, $P(\Gamma_r=0/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت") is a parasite probability because the word النوعية is not an ingredient in reality given الجيدة have an ingredient state $\Gamma_r=0$. 22
23 situations table for $\delta_5(7,2)+log(a_{721})$ for a double agent Viterbi with external knowledge transfer, after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, $P(\Gamma_r=0/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت") is not a parasite probability because the word النوعية is not an ingredient and النوعية POS tag is منعوت 23
24 situations table for $\delta_5(7,3)+log(a_{731})$ for a double agent Viterbi with external knowledge transfer, after adding the sentence كعبة طماطم من الحجم الكبير to the dataset, $P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت") is a parasite probability because the word النوعية have an ingredient state $\Gamma_{r-1}=0$ and not $\Gamma_{r-1}=1$. . . 23
25 situations table for $\delta_5(7,1)+log(a_{711})$ for a double agent Viterbi with external knowledge transfer after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, $P(\Gamma_r=0/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت") is not a parasite probability because the word النوعية is not an ingredient and the word النوعية is منعوت given the word الجيدة is not an ingredient. 24
26 situations table for $\delta_5(7,2)+log(a_{721})$ for a double agent Viterbi with external knowledge transfer after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, $P(\Gamma_r=1/\Gamma_{r-1}=0, T_{r-1}=$ "منعوت") is not a parasite probability because the word النوعية is not an ingredient and the word النوعية is منعوت given the word الجيدة is not an ingredient. 25
27 situations table for $\delta_5(7,3)+log(a_{731})$ for a double agent Viterbi with external knowledge transfer after adding the sentence زيت زيتون من النوعية الجيدة to the dataset, $P(\Gamma_r=2/\Gamma_{r-1}=1, T_{r-1}=$ "منعوت") is a parasite probability because the word النوعية is not an ingredient in reality. 25

28 Performances for each tag of a first order HMM , the trained dataset is used as a test dataset , and there is zero unknown words to predict 27
29 Performances for each tag of a second order HMM , the trained dataset is used as a test dataset , and there is zero unknown words to predict 28
30 Performance comparison using a simple double agent Viterbi with external knowledge transfer with 100 % accuracy on first layer , the trained dataset is used as a test dataset to avoid unknown words prediction 29
31 Comparison of performances between a simple double agent Viterbi algorithm with external knowledge transfer and other HMM mono-agent methods using the trained corpus as a testing dataset . . . 31
32 Performances of first order HMM used to predict ingredients state using 10 fold cross-validation with unknown word consideration, the test dataset have 20 % sentences from the total dataset . . . 31
33 Performances of second order HMM used to predict ingredients state using 10 fold cross-validation with unknown word consideration, the test dataset have 20 % sentences from the total dataset 32
34 Average performances in predicting ingredients states with $\lambda = 4$ when accuracy is 100% on first layer with our ingredient extractor 32
35 Average performances in predicting ingredients states with $\lambda = 4$ when accuracy is 100% on first layer using a double layered Viterbi Algorithm 33
36 Comparison of performances between the 4 methods using 20 % of the total dataset as a test dataset 33
37 situations table for $\delta_5(12, 1) + log(a_{1211})$ for a simple double agent Viterbi without unknown words consideration, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} =$ "إسم مجرور") is a parasite probability because the real POS tag for the word جافة is نعت and not إسم مجرور and the ingredient state of the word جافة is $\Gamma_{r-1} = 2$ and not $\Gamma_{r-1} = 1$ 35
38 situations table for $\delta_5(12, 2) + log(a_{1221})$ for a simple double agent Viterbi without unknown words consideration, the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 3, T_{r-1} =$ "إسم مجرور") is a parasite probability because the real POS tag for the word جافة is نعت and not إسم مجرور and the ingredient state of the word جافة is $\Gamma_{r-1} = 2$ and not $\Gamma_{r-1} = 3$ 35
39 Situations table for $\delta_5(12, 1) + log(a_{1211})$ for a simple double agent Viterbi with external knowledge transfer without unknown words consideration, the parasite probability probability $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} =$ "إسم مجرور") is eliminated and replaced by the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} =$ "نعت") 36

40 Situations table for $\delta_5(12,2)+log(a_{1221})$ for a simple double agent Viterbi with external knowledge transfer without unknown words consideration, the parasite probability probability $P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} =$ "إسم مجرور") is eliminated and replaced by the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 2, T_{r-1} =$ "نعت") 37
41 Situations table for $\delta_5(9,1)+log(a_{911})$ for a simple double agent Viterbi with unknown words consideration without external knowledge transfer, $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} =$ "إسم مجرور") is a parasite probability because the real ingredient state of the word جافة is $\Gamma_{r-1} = 2$ and the real POS tag of the word جافة is نعت . 38
42 Situations table for $\delta_5(9,2)+log(a_{921})$ for a simple double agent Viterbi with unknown words consideration, $P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} =$ "إسم مجرور") is a parasite probability because the real ingredient state of the word جافة is $\Gamma_{r-1} = 2$ and the real POS tag of the word جافة is نعت 39
43 Situations table for $\delta_5(9,1)+log(a_{911})$ for a simple double agent Viterbi with external knowledge transfer with unknown words consideration, the parasite probability probability $P(\Gamma_r = 0/\Gamma_{r-1} = 1, T_{r-1} =$ "إسم مجرور") is eliminated and replaced by the probability $P(\Gamma_r = 0/\Gamma_{r-1} = 2, T_{r-1} =$ "نعت") 39
44 Situations table for $\delta_5(9,2) + log(a_{921})$ for a primitive double agent Viterbi with external knowledge transfer with unknown words consideration, the parasite probability probability $P(\Gamma_r = 1/\Gamma_{r-1} = 3, T_{r-1} =$ "إسم مجرور") is eliminated and replaced by the probability $P(\Gamma_r = 1/\Gamma_{r-1} = 2, T_{r-1} =$ "نعت") 40

References

Baklouti, Z. (2021, September). *Hidden Markov Based Mathematical Model dedicated to Extract Ingredients from Recipe Text.* Retrieved from `https://hal.archives-ouvertes.fr/hal-03355929` (working paper or preprint)

Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1998). Nymble: a high-performance learning name-finder. *arXiv preprint cmp-lg/9803003*.

Freitag, D., & McCallum, A. (1999). Information extraction with hmms and shrinkage. In *Proceedings of the aaai-99 workshop on machine learning for information extraction* (pp. 31–36).

Freitag, D., & McCallum, A. (2000). Information extraction with hmm structures learned by stochastic optimization. *AAAI/IAAI*, *2000*, 584–589.

Leek, T. R. (1997). *Information extraction using hidden markov models* (Unpublished master's thesis). Citeseer.

Morwal, S., Jahan, N., & Chopra, D. (2012). Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol*, *1*.

Seymore, K., McCallum, A., Rosenfeld, R., et al. (1999). Learning hidden markov model structure for information extraction. In *Aaai-99 workshop on machine learning for information extraction* (pp. 37–42).

Shah, L., Gaudani, H., & Balani, P. (2016). Personalized recipe recommendation system using hybrid approach. *International Journal of Advanced Research in Computer and Communication Engineering*, 192–197.

Ueda, M., Asanuma, S., Miyawaki, Y., & Nakajima, S. (2014). Recipe recommendation method by considering the users preference and ingredient quantity of target recipe. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol. 1, pp. 12–14).

Zhu, Y., Chen, Y., Lu, Z., Pan, S. J., Xue, G.-R., Yu, Y., & Yang, Q. (2011). Heterogeneous transfer learning for image classification. In *Twenty-fifth aaai conference on artificial intelligence.*