

Assistenti virtuali e Natural Language Processing

Andrea Broccoletti

Istituto di Istruzione Secondaria Superiore
"Ettore Majorana" - Seriate

Abstract- Il machine learning è ormai alla base di svariati aspetti della società moderna: dalle ricerche sul web al filtraggio dei contenuti, dal riconoscimento di immagini alla comprensione della lingua parlata. Tutte queste applicazioni fanno uso di algoritmi e tecniche che appartengono ad una branca dell'intelligenza artificiale chiamata deep learning. Grazie alle reti neurali artificiali in grado di analizzare automaticamente i dati, questo ambito sta vivendo anni di rapido progresso arrivando anche, in alcuni casi, a superare le prestazioni degli esseri umani. Il Natural Language Processing fa parte di questo ambito, in quanto alle sue fondamenta vi sono deep learning e reti neurali. Questo studio si concentra proprio sull'NLP, in particolare la sua applicazione negli assistenti virtuali commerciali quali, per citare i più diffusi [1], Google Assistant, Alexa, Siri e Cortana.

Parole chiave- Assistente virtuale, Natural Language Processing, rete neurale.

I. INTRODUZIONE

Gli assistenti virtuali sono oramai dei software comuni ed usati quotidianamente, il cui principale obiettivo è quello di replicare le modalità con cui l'uomo comprende i concetti al fine di affrontare una comunicazione vera e propria con un interlocutore umano [2]. Il termine è usato generalmente per indicare qualsiasi software in grado di dialogare, compresi quelli che utilizzano schemi logici preimpostati di domanda/risposta, le chatbot, che rappresentano il livello fondamentale di qualsiasi assistente virtuale. L'interazione tra le tecnologie alla base delle chatbot, il deep learning e le reti neurali ha portato alla nascita di assistenti virtuali in grado di interpretare il linguaggio umano e rispondere attraverso voci sintetizzate, che simulano il linguaggio e il parlato umano e rendono completamente trasparente la natura artificiale degli assistenti, fornendo all'utilizzatore un'esperienza del tutto equiparabile ad un dialogo con un individuo umano.

Il Natural Language Processing (NLP) è proprio alla base di queste tecnologie, ed è una branca dell'intelligenza artificiale che si occupa della progettazione di sistemi in grado di analizzare e comprendere il linguaggio umano e di produrre risposte logiche in linguaggio naturale. [3].

II. TECNICHE ALLA BASE DELL'NLP

A differenza di grande parte delle informazioni elaborate dalle macchine, che seguono regole ben precise e sono facilmente interpretabili dalle macchine, la lingua da noi utilizzata non è facilmente rappresentabile [4]. È necessario quindi costruire sistemi in grado di comprendere e rispondere all'uomo. Ed è qui che entrano in gioco i sistemi di intelligenza artificiale, in particolare sono due le tecniche alla base dell'NLP:

a. Deep Learning

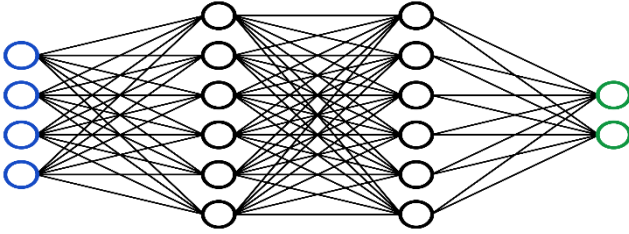
È una sottocategoria del machine learning, il cui fine è generare modelli di apprendimento, è quindi corretto definirlo come l'apprendimento di dati che non sono forniti dall'uomo alla macchina, ma sono appresi grazie all'utilizzo di algoritmi di calcolo statistico. Il deep learning tenta in tutto e per tutto di imitare il modo in cui il nostro cervello raccoglie e elabora informazioni [5]: un uomo non impara ricevendo istruzioni e set di dati, ma riceve dei modesti dati in input che, elaborati, producono informazioni e conoscenza.

b. Reti Neurali Artificiali

Per arrivare a comprendere cosa è una rete neurale, è doveroso introdurre cosa è una rete neurale biologica: un circuito formato da neuroni che svolgono una funzione di riconoscimento, memorizzazione e reazione agli stimoli provenienti dall'esterno; sono come delle strade che vengono costruite pian piano, all'inizio non portano da nessuna parte ma, con il tempo e con l'analisi di una grande mole di stimoli, riusciamo a costruire un reticolo di autostrade che percorriamo ogni qual volta dobbiamo ragionare, fare calcoli, riconoscere immagini, imparare e agire [6].

Le reti neurali artificiali non fanno altro che copiare questo sistema: sono modelli matematici composti da neuroni artificiali che vengono utilizzati per permettere ad una macchina di apprendere. L'apprendimento può avvenire in svariati modi, ma quello che interessa gli assistenti virtuali è solo uno ed è chiamato apprendimento supervisionato [7], si fornisce quindi alla rete il training set, ovvero un insieme di input ai quali corrispondono output noti (nel caso degli assistenti virtuali, si forniscono alla rete numerosi esempi di un comando, e quale deve essere la risposta dell'assistente virtuale al comando in esame), la

rete, analizzandoli, apprende il nesso che li unisce e in tal modo impara a generalizzare, ovvero a calcolare nuove associazioni input-output che non siano state presenti nel training set.



Queste due tecniche non sono solo alla base dell’NLP, sono molto di più, sono le tecniche che umanizzano gli assistenti virtuali: il deep learning non è altro che il ragionamento, il pensiero, la conoscenza, mentre la rete neurale è il cervello della macchina, senza il quale non saprebbe ragionare.

III. IL NATURAL LANGUAGE PROCESSING

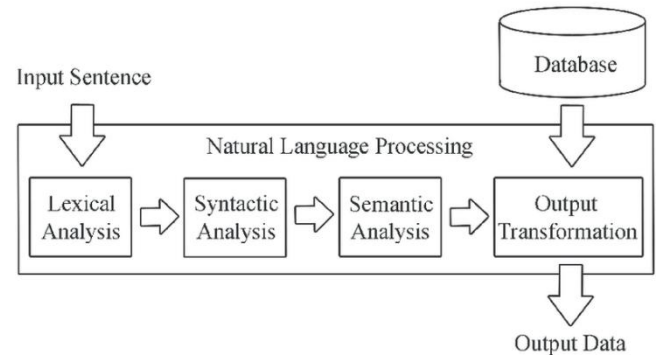
Affrontate le tecniche alla base dell’NLP, si può dare una definizione più rigorosa [4]:

Per natural language processing si intendono algoritmi di intelligenza artificiale in grado di analizzare, rappresentare e quindi comprendere il linguaggio naturale.

L’NLP si occupa principalmente di testi, intesi come sequenze di parole che in una lingua esprimono uno o più messaggi, mentre l’elaborazione del parlato è considerato un ambito a sé.

Il dialogo tra uomo e macchina coinvolge diversi aspetti come fonetica, morfologia, sintassi, semantica e il discorso nel suo complesso. Di conseguenza, bisogna vedere l’NLP come un insieme di fasi successive e dipendenti molto differenti tra loro, che riguardano quindi l’analisi di diversi aspetti del linguaggio naturale [9]:

- *Pre-analisi*: consiste nell’eliminare spazi, interpunzioni e punteggiatura, e parole senza significato
- *Analisi lessicale*: attraverso lo studio e la scomposizione delle frasi in input vengono identificate le parole chiave
- *Analisi sintattica*: utilizzata per capire le relazioni fra le singole unità della frase
- *Analisi semantica*: consiste nell’identificare delle entità con significato semantico



Questa successione di analisi è la stessa che noi applichiamo per comprendere una frase, quindi, come già capitato con il deep learning e le reti neurali, anche l’NLP simula il modo di elaborare il linguaggio che utilizziamo noi umani.

a. Pre-analisi

È la fase preliminare, e a sua volta si compone di tre sottofasi:

- *Pulizia dell’input*: la frase da analizzare, che viene vista come una stringa di caratteri, viene ripulita di ogni carattere non funzionale alle successive analisi, quindi tabulazioni, spazi e punteggiatura vengono eliminati dalla frase;
- *Suddivisione*: in parallelo alla sottofase precedente, ogni set di caratteri suddiviso da uno spazio viene identificato come una singola entità, la parola
- *Identificazione dei lessemi*: vengono identificate le parole che sono lessemi, ovvero minime unità linguistiche aventi un significato autonomo, e vengono scartate tutte le parole che non saranno utili alle successive analisi.

b. Analisi Lessicale

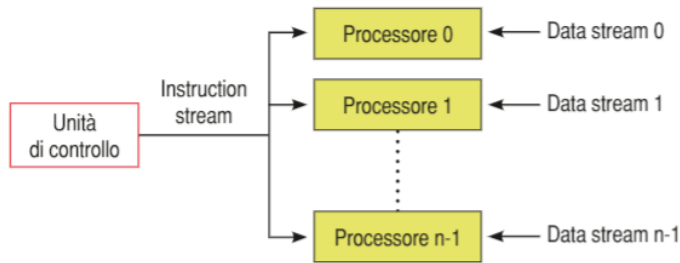
È il processo attraverso il quale si comprende il significato delle parole, intuendo il loro contesto. Operativamente, consiste nel tradurre una parola in una stringa di caratteri che avrà un determinato valore lessicale, chiamato token. Successivamente, si procede alla ricerca delle parole in un dizionario per estrarne il significato. In questa fase è importante notare, soprattutto nella lingua italiana, che una stessa parola può assumere diversi significati, a cui corrispondono diversi valori lessicali, ma di questo se ne occuperà la seconda fase di analisi.

Il tipo di token identifica il ruolo che la parola, ora chiamata valore del token, ha nella frase, per esempio la frase “meteo a Bergamo”, suddivisa in token, diventerà:

token	valore
METEO	Literal
BERGAMO	Identifier

descritti nella sezione precedente. Quindi è necessario avere un'infrastruttura hardware in grado di soddisfare la richiesta. Attualmente, il mercato del server hosting dispone di hardware appositamente progettato per rispondere alle esigenze del deep learning, come i server OVH [15], o addirittura di server progettati per il natural language processing, come i Cloud Natural Storage di Google [16]. Indipendentemente dal servizio di hosting scelto o dalle funzionalità, alla base di tutto c'è un importante componente hardware: la Graphic Processing Unit (GPU) [17].

Può risultare paradossale usare la GPU, ideata per i calcoli che riguardano la grafica, e non la CPU, ma la ragione è molto semplice: la GPU ha molte unità logiche aritmetiche, dei processori, il cui scopo di base è quello di elaborare una serie di calcoli semplici e identici; questo tipo di architettura viene chiamata Single Instruction Multiple Data, poiché più processori eseguono la stessa istruzione su flussi di dati diversi in parallelo [25].



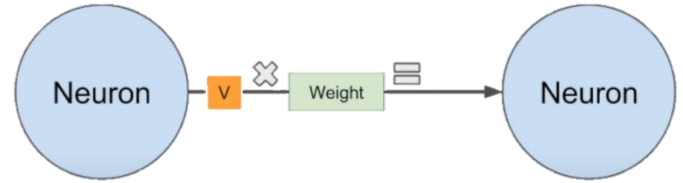
La GPU ha questa architettura perché i calcoli per la grafica sono di tipo matriciale e vettoriale, e quindi richiedono potenza di calcolo “in parallelo”. Questo calcolo parallelo è richiesto anche dal natural language processing, alla cui base c'è ancora il calcolo matriciale e vettoriale [18], risulta quindi evidente perché i server necessitano di molta GPU: perché sono ottimizzate per effettuare più calcoli contemporaneamente.

VI. FONDAMENTI MATEMATICI

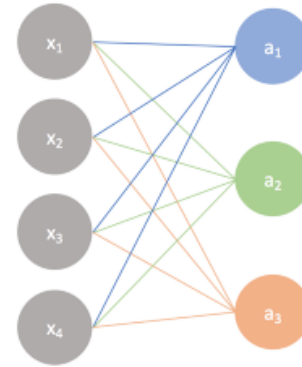
Nei requisiti hardware, si è compreso che alla base del deep learning e delle reti neurali e, di conseguenza, anche del natural language processing, c'è il calcolo matriciale. Ovviamente i fondamenti matematici nel “dietro le quinte” di queste tecnologie sono molti di più e molto più complessi, ma al fine di questo studio è necessario solo comprendere il calcolo matriciale applicato alle reti neurali. Per qualsiasi rete neurale, la fase di addestramento del modello di deep learning è l'attività che richiede più risorse. Durante l'addestramento, una rete neurale riceve input, che vengono quindi elaborati in livelli nascosti utilizzando pesi che vengono regolati durante l'addestramento e il modello emette quindi una previsione. I pesi vengono regolati per trovare modelli al fine di fare previsioni migliori.

Entrambe queste operazioni sono essenzialmente moltiplicazioni matriciali [17].

L'operazione più comune in una rete neurale è il calcolo dei pesi e del valore di un nuovo neurone. Un nuovo neurone, ha come valore il prodotto tra il peso di un collegamento e il valore in uscita del neurone all'altro capo del collegamento:



Considerando che un neurone è collegato, solitamente, a più di un neurone, e considerando questa rete neurale estremamente semplificata:



Il calcolo matriciale per calcolare il valore in ingresso dei nuovi neuroni sarà:

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \rightarrow \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Questo tipo di calcolo, viene effettuato in parallelo un numero considerevole di volte, sia in fase di training sia durante l'utilizzo della rete neurale, del natural language processing e, quindi, ogni volta che effettuiamo una richiesta al nostro assistente virtuale.

VII. CASO D'USO: AV E NLP IN PYTHON

Il natural language processing, per quanto possa sembrare una tecnologia avanzata e ancora in via di sviluppo, è invece già sviluppato e non solo da un punto di vista industriale.

Infatti, come dimostrato nella sezione precedente, se da un lato i fondamenti teorici e matematici che sono alla sua base sono complessi [19], dall'altro la community di

programmatore è già da molti anni impegnata ad offrire librerie, cioè collezioni di funzioni utilizzabili da altri programmi [20], che consentono ai programmatori di astrarre dalle proprie capacità matematiche, concentrandosi solo sull'integrazione di queste librerie nei propri codici.

Il linguaggio di programmazione più utilizzato in questo campo è Python, perché offre un codice conciso e leggibile; sebbene gli algoritmi di natural language processing siano complessi, la semplicità di Python consente agli sviluppatori di scrivere codice comprensibile e scalabile [21].

La sua semplicità, unita alle avanzate librerie disponibili, consente la creazione di un assistente virtuale affidabile, non al pari di quelli commerciali per evidenti mancanze di risorse, ma del tutto utilizzabili e migliorabili.

In particolare, il caso d'uso successivamente rappresentato farà uso delle seguenti librerie:

- *Speech recognition by Google* [22], per la sintesi vocale
- *Nltk* [10], per il natural language processing
- *Tensorflow* [23], per la rete neurale e le librerie di machine learning necessarie
- *Pytsx3* [24] per la sintesi vocale

Il caso d'uso seguente è il codice sorgente, in linguaggio Python, di un assistente virtuale essenziale, che è in grado di comunicare data e ora, previsioni del tempo, presentarsi e salutare. Nell'esposizione del codice, si potranno ritrovare tutte le sezioni teoriche precedenti, presentate stavolta sotto il punto di vista tecnico-operativo. La fase di training della rete neurale è già stata eseguita e non farà parte del codice seguente.

a. Voice recognition

Grazie alla libreria SpeechRecognition e alle API di Google, l'implementazione di questa prima fase è relativamente semplice:

```
listener.dynamic_energy_threshold = False
listener.energy_threshold = 400
voice = listener.listen(source, timeout=3)
awake = listener.recognize_google(voice, language="it-IT")
awake = awake.lower()
```

Viene impostata una soglia di ascolto di 400/1000, per evitare che i rumori ambientali possano influire negativamente sull'ascolto, successivamente con due semplici richiami alla libreria avviene la conversione da parlato a stringa testuale.

b. Natural Language Processing

Questa fase sfrutta la rete neurale generata precedentemente. Fa utilizzo di una serie di funzioni che

svolgono, rispettivamente, le fasi da cui l'NLP è composto, ricevendo la stringa generata dalla fase precedente e restituendo il significato, ovvero l'intent, della richiesta.

```
# pulizia dell'input e divisione in token della stringa
def clean_up_sentence(sentence):
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words = [lemmatizer.lemmatize(word) for word
                      in sentence_words]
    return sentence_words

# suddividere la stringa in token e li numera
def bag_of_words(sentence):
    sentence_words = clean_up_sentence(sentence)
    bag = [0] * len(words)
    for w in sentence_words:
        for i, word in enumerate(words):
            if word == w:
                bag[i] = 1
    return np.array(bag)

# ricerca ogni token nella rete neurale per
# comprendere il significato dei token e della frase
def predict_class(sentence):
    bow = bag_of_words(sentence)
    res = model.predict(np.array([bow]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i, r] for i, r in enumerate(res)
               if r > ERROR_THRESHOLD]
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({'intent': classes[r[0]],
                           'propability':str(r[1])})
    return return_list

# data una stringa in input, restituisce il suo significato
# definito nella rete neurale precedentemente trainata
def get_response(intents_list, intenst_json):
    tag = intents_list[0]['intent']
    list_of_intents = intenst_json['intents']
    for i in list_of_intents:
        if i['tag'] == tag:
            result = random.choice(i['responses'])
            break
    return result
```

c. *Information Retrieval*

Il Knowledge base può essere implementato in diversi modi, gran parte dei quali prevedono l'utilizzo di un database. Dato che l'assistente virtuale in questione è essenziale e con un set di funzioni limitato, l'implementazione di questa fase è stata effettuata con un semplice costrutto `if...elif...else`:

```
if res != None:
    if res == "time_engine":
        talk(get_time())
    elif res == "meteo_engine":
        talk(get_wheater(command))
    elif res == "date_engine":
        talk(get_date())
    else:
        talk(res)
```

Ogni funzione dell'assistente virtuale, corrisponde ad un ramo del costrutto. In base al risultato ottenuto dalla fase precedente (`res`), verrà intrapresa l'azione corrispondente.

d. *Speech synthesis*

A questo punto, non rimane altro che fornire l'output, come risposta all'utente interlocutore. La Speech Synthesis avviene in due fasi. Nella prima si definiscono le caratteristiche della voce che verrà utilizzata:

```
voiceRate = 140
engine = pyttsx3.init()
engine.setProperty('rate', voiceRate)
```

Nella seconda si definisce una funzione che sarà utilizzata ogni volta che l'assistente virtuale dovrà parlare, ovvero ogni volta che un risultato, in formato di stringa, dovrà essere convertito in voce:

```
def talk(sentence):
    engine.say(sentence)
    engine.runAndWait()
```

Il risultato di tutte queste sezioni di codice è un'assistente virtuale funzionante, basato su una rete neurale e con un motore NLP completo.

VIII. CONCLUSIONI

Gli assistenti virtuali sono parte delle nostre vite, dallo smartphone fino al servizio clienti di un'azienda. Possono essere elaborati e complessi e risiedere in un server in qualche server farm, o possono essere semplici e programmabili interamente dal computer di casa.

Utilizzano il natural language processing per comprenderci e per togliere l'ultima barriera disumanizzante che li separa da noi. Tecnologia, pura tecnologia al nostro servizio, che sfrutta la matematica per essere non solo artificiale, ma anche intelligente.

IX. SITOGRAFIA

- [1] **Voicebot.ai**, *45% of Millennials Use Voice Assistants While Shopping According to a New Study* [https://voicebot.ai/2019/03/20/45-of-millennials-use-voice-assistants-while-shopping-according-to-a-new-study]
- [2] **Quest-it.com**, *Assistente virtuale o chatot? La differenza la fa l'intelligenza artificiale* [https://www.quest-it.com/blog/assistente-virtuale-o-chatbot-la-differenza-la-fa-lintelligenza-artificiale]
- [3] **Treccani.it**, *Definizione di linguaggio naturale* [https://www.treccani.it/enciclopedia/linguaggio-naturale_%28Enciclopedia-della-Matematica%29/#:~:text=La%20scrittura%20e%20o%20studio,di%20grammatica%20generatrice%20del%20linguaggio]
- [4] **Osservatori.net**, *Natural language Processing (NLP): come funziona l'elaborazione del linguaggio naturale* [https://blog.osservatori.net/it_it/natural-language-processing-nlp-come-funziona-l-elaborazione-del-linguaggio-naturale]
- [5] **Intelligenzaartificiale.it**, *Deep Learning* [https://www.intelligenzaartificiale.it/deep-learning/]
- [6] **Ai4business.it**, *Reti neurali: cosa sono e a cosa servono* [https://www.ai4business.it/intelligenza-artificiale/deep-learning/reti-neurali/]
- [7] **Intelligenzaartificiale.it**, *reti neurali* [https://www.intelligenzaartificiale.it/reti-neurali/]
- [8] **Atgartificialintelligence.com**, *nlp cosa è e come può aiutarti ad essere più efficiente* [https://atgartificialintelligence.com/nlp-cose-e-come-puo-aiutarti-ad-essere-piu-efficiente/#:~:text=L'elaborazione%20del%20linguaggio%20naturale,computer%20e%20il%20linguaggio%20umano]
- [9] **@CKESpanol attraverso medium.com**, *What are the different levels of NLP?* [https://medium.com/@CKESpanol/what-are-the-different-levels-of-nlp-how-do-these-integrate-with-information-retrieval-c0de6b9ebf61]
- [10] **Nltk.org**, *natural Lanuage Toolkit* [https://www.nltk.org]
- [11] **Tandfonline.com**, *An introduction to voice assistants* [https://www.tandfonline.com/doi/abs/10.1080/02763869.2018.1404391?journalCode=wmrs20]
- [12] **G2.com**, *What is a voice assistant and are they the future chatbots* [https://www.g2.com/articles/voice-assistant]
- [13] **Amazon.it**, *A casa con Alexa* [https://www.amazon.it/adlp/acasaconalexa]
- [14] **Ccn.com**, *"I'm the original voice of siri"* [https://edition.cnn.com/2013/10/04/tech/mobile/ben-nett-siri-iphone-voice/index.html]
- [15] **Ovhcloud.com**, *Ovhcloud dedicated servers for AI and machine learning* [https://ovhcloud.com]
- [16] **Google.com**, *Natural Language* [https://cloud.google.com/natural-language]
- [17] **Towardsdatascience.com**, *What is a gpu and do you need one in deep learning?* [https://towardsdatascience.com/what-is-a-gpu-and-do-you-need-one-in-deep-learning-718b9597aa0d]
- [18] **Oracle.com**, *AI and Datascience* [https://blogs.oracle.com/ai-and-datascience/post/cpu-vs-gpu-in-machine-learning]
- [19] **Souman Roy attraverso medium.com**, *Math, Stats and NLP for Machine Learning: As Fast As Possible* [https://medium.com/meta-design-ideas/math-stats-and-nlp-for-machine-learning-as-fast-as-possible-915ef47ced5f]
- [20] **Treccani.it**, *Definizione di libreria* [https://www.treccani.it/enciclopedia/libreria/#:~:text=libreria%20In%20informatica%2C%20collezione%20di,le%20procedure%20di%20una%20l.]
- [21] **Steelkiwi.com**, *Why use Python for AI and machine learning?* [https://steelkiwi.com/blog/python-for-ai-and-machine-learning/#:~:text=Python%20offers%20concise%20and%20readable,developers%20to%20write%20reliable%20systems.&text=Python%20code%20is%20understandable%20by,build%20models%20for%20machine%20learning.]
- [22] **Pypi.org**, *SpeechRecognition* [https://pypi.org/project/SpeechRecognition/]
- [23] **Tensorflow.org**, *Tensorflow* [https://www.tensorflow.org/]
- [24] **Pypi.org**, *Pytsx3* [https://pypi.org/project/pytsx3]

X. BIBLIOGRAFIA

- [25] **Hoepli**, *Tecnologie e progettazione di sistemi informatici e di telecomunicazioni* [ISBN 9788820378424]